NatQuery

Installation and Configuration Manual

Table of Contents

Introduction

NatWorks, Inc. is very pleased to have you as a customer of NatQuery, and we sincerely hope that you find NatQuery easy to install, configure, and use.

This document applies to NatQuery Version 6.x and all subsequent versions, unless indicated in new editions, technical newsletters or release notes. Specifications contained herein are subject to change, and these changes will be appropriately reported in subsequent Release Notes and / or new editions of this document.

Intended Audience

This document is designed to be utilized by an individual who is or will be responsible for the installation, configuration and maintenance of NatQuery, and therefore the content is geared towards the topics that surround this work. Throughout this manual, this individual will be referred to as the NatQuery Administrator.

Overview of NatQuery

Simplistically, NatQuery is a workstation-based application that automates data extraction and auditing against for Software AG's ADABAS database.

Specifically, NatQuery provides a graphical-user interface that allows for the generation of processing that accomplishes extraction of ADABAS data, such as:

- NatQuery for End-User Data Extraction,
- NatQuery for Data Warehousing, and
- NatQuery for Processing Data into ADABAS

Common examples of the type of data handling NatQuery can perform would be:

- End-User Adhoc Query of ADABAS Data with integration to Desktop Tools such as Microsoft Excel or Access
- Integration of ADABAS Data to tools such as IBM / Ascential Software's DataStage Suite, Pervasive Software's Data Integrator and other similar tools
- Direct integration of ADABAS Data to popular RDBMS such as Microsoft SQL Server, Oracle and MySQL.
- Data extraction from one ADABAS environment for automated movement and loading into a separate ADABAS environment.
- Data extraction and conversion of ADABAS data into transfer formats such XML (with optional XSL), and delimited or non-delimited support.

When combined with NatCDC (NatWorks Protection Log Processor), NatQuery can additionally generate Natural programs that allow for the auditing of ADABAS transactional information; this functionality is discussed in greater detail in the manual **Auditing ADABAS Protection Logs with NatQuery**, which is provided separately.

NatQuery achieves these goals by intelligently generating Natural programs within the workstation environment according to user supplied specifications. In addition to generating these programs, NatQuery can:

- Generate objects such as Job Control Language (JCL) or Scripts that allow for the execution of the generated processing in a Batch environment on the Natural Server platform;

- Introduce these objects onto the Natural server platform for automatic execution;
- Allow the results of completed data extraction processes to be retrieved; and
- Automatically handle all ADABAS data formats, with the ability to integrate this data into a wide-variety of targets.

To achieve automated movement of generated processes / objects between NatQuery in a workstation environment and Natural server platform, NatQuery primarily supports the use of automated File Transfer Protocol (FTP) or FTPS over TCP/IP to achieve integration to the Natural / ADABAS server platform, or can alternatively utilize automated File Copy operations across a network where this is supported.

To achieve automatic execution of extracts, several methods are supported depending on the Natural server platform. If the Natural server platform is MVS or VSE, then automated FTP / FTPS operations can be performed that move generated JCL into files on the Natural server platform for later manual execution. Beyond moving these files, FTP / FTPS can be configured in mainframe environments such that FTP / FTPS operations place JCL directly into JES (MVS systems) or POWER (VSE systems); thus achieving automatic execution in an easily controlled fashion.

If the Natural server platform is UNIX / Linux or Windows, then automated execution can be accomplished via one of several Remote Execution approaches such as RSH, REXEC, PSEXEC, or PLINK (a component of PuTTY).

Against all Natural targets, and in situations where Remote Execution solutions are not otherwise viable or allowed, server based processing that “polls and submits” can be used to achieve automated execution. These types of execution solutions are available from third parties, through site-specific “home-grown” solutions, or in some cases, examples can be made available directly from NatWorks.

While many methods exist to automatically move and automatically execute NatQuery-generated extraction requests, there is no requirement that generated processes / objects must be either automatically executed on the Natural server, or even automatically moved to the Natural server.

As all NatQuery requests physically execute in a Batch environment, a NatQuery installation provides a set of generic JCL / Script templates with each of these templates being specific to a given task that NatQuery may be asked to perform. Through configuration, some or all of these templates are customized so that they will operate as intended within a customer’s specific environment.

In almost all cases, these templates will contain references to “Dynamic Substitution Variables” (uniquely tagged fields), with NatQuery replacing these Dynamic Substitution Variables with appropriate values when an execution request is processed. In this way, a “generic” template is customized to be a fully executable process that will process the users’ specific request(s).

To track submitted requests, NatQuery supports a simple logging mechanism that allows the status of any submitted request to be monitored. When a user submits a request to the Natural server, NatQuery assigns it a unique request number, and then stores information related to the request to a text file within the workstation environment that is specific to the submitting user (a user’s “**Local Log**” file). A user’s **Local Log** file is used to track what requests were submitted for execution, as well as providing the next unique Request Number for the user. When a specific user’s request executes on the Natural server, then in addition to writing out the requested data into an output file, the request process will also typically update a server-based text file that is specific to the submitting user (a user’s “**Remote Log**” file). A user’s **Remote Log** file then records the execution status of submitted requests.

With the existence of these two files (a user’s **Local Log** of what was sent and **Remote Log** of what has been executed), NatQuery can retrieve the **Remote Log** and then programmatically compare the respective contents to the **Local Log,** thus providing a simple and straightforward mechanism of allowing a user to monitor the status of any specific request. When monitoring indicates that any given request is “**Done**”, the user can then initiate an automated download of the resulting data to bring that data into the user's workstation environment.

Once the user of NatQuery retrieves the output of an extraction request, further automatic processing may then occur, depending on the user-designated target. For example, if data was requested to be placed into MS Excel or MS Access – this integration can be automatically performed subsequent to that data being downloaded.

NatQuery for End-User Data Extraction

From its initial conception, NatQuery was primarily designed to be an End-User tool, a tool that will allow an End-User to access ADABAS data in an intelligent and controllable manner.

An End-User is empowered to request data extracts against one or more ADABAS files, and can then automatically have this data placed into a number of targets such as:

- Microsoft Excel
- Microsoft Access
- XML (with optional XSL)
- DB2
- SQL Server
- Oracle
- MySQL
- Text Files with fixed-length fields, either delimited or non-delimited

Through an easy-to-understand and easy-to-use Graphical User Interface (GUI), an End-User is able to leverage the knowledge / information contained in an administratively built **Environment Configuration**. Through this interface, an End-User can create requests for data extraction (henceforth referred to as **Query Specifications**). In most cases the user can then additionally submit these requests for execution; they can monitor the execution status of these submitted requests; they can automatically retrieve the extracted data when the requests shows as complete, and they can then automatically integrate this data into the targets listed above.

To assist the End-User, **Query Specifications** can be saved into a user-specified directory (either local to the NatQuery installation or into a network location), and when these are saved the user can attach both short and long textual descriptions, as well as having the ability to mark specific queries as "favorites", thus allowing for the easy identification and retrieval for re-execution and / or modification and execution.

Query Specifications can be handled in typical Windows fashion with functions of **Open**, **Save**, **Save As**, **Delete**, **Import** and **Export**.

When handling "recurring" fields (field structures that are unique to ADABAS such as Multi-Valued Fields (MUs) or Periodic-Groups (PEs)), these fields are presented to the user as a single field that is “tagged” to show that it recurs, with the tag reflecting the administratively defined occurrence default(s). For example; a recurring Multi-Valued (MU) field named ADDRESS-LINE could be referenced with syntax similar to ADDRESS_LINE(ALL), or a MU field called BONUS that existed in a periodic-group could be referenced with BONUS(ALL,ALL) – with the “ALL” value representing a administratively defined maximum number of occurrences that can exist to be handled. This feature greatly reduces the level of knowledge that End-Users must have to utilize these recurring fields effectively, as a single field selection will have the effect of returning all individual occurrences. Alternatively, a user can select a single occurrence, or a range of occurrences, to be extracted - as long as these occurrences are within the Administratively defined maximums.

When outputting recurring fields, End-Users have the option of having these fields “strung together” (I.E. “concatenated”) or “normalized” (I.E. “flattened”).

To extend the fields available for extraction beyond the base ADABAS fields themselves, an End-User has the option to create “variables”. Examples of NatQuery-supported variables include **Redefinitions**, **Constants**, **Expressions**, **Compressions**, and the use of **Dynamic Variables** (Date and Timestamp values as well as User-Input values).

To insure that End-Users do not release extraction requests that could negatively impact the source system, the Administrator can place restrictions / constraints on individual users. The Administer can control such things as what I/O statements a user can utilize, as well as how many records a given query can read before it will be automatically terminated.

In achieving extraction, NatQuery natively embraces all levels of security in place at a customer site, including system-level security, Natural Security and even ADABAS Security.

NatQuery for Data Warehousing

One of the most difficult tasks with any Data Warehousing effort is the implementation and management of the physical data capture processes. The following topics relate how NatQuery can assist with this effort:

- Data Extraction
- Change Data Capture
- Integration to Extraction Transformation and Loading (ETL) Tools
- Stand-Alone Data Warehouse Support

Data Extraction

For general Data Extraction, NatQuery is all that is required. For maximum flexibility, NatQuery allows for the generation of Natural data extraction programs that will achieve extraction all by themselves. For maximum performance, NatQuery supports the generation of the parameters required for specific ADABAS utilities, as well as the generation of Natural programs that process the data that these utilities produce. In current versions of NatQuery, the ADABAS unload utilities that are supported are ADAULD and ADACMP.

Change Data Capture

For Change Data Capture, NatWorks offers the product NatCDC, an add-on product to NatQuery. Through the use of NatCDC, an organization can easily generate complete processes that will successfully process the ADABAS transactional data stored in ADABAS Protection Logs (PLOGS). The ability to process ADABAS PLOGS allows for the ability to support Change Data Capture (CDC) processing as well as enabling full transaction auditing.

To support PLOG processing, NatCDC makes use of the ADABAS utilities that Software AG provides with ADABAS to "pre-process" PLOG data. The final processing is provided by NatQuery generated processes that will utilize NatQuery-generated Natural Programs. When ADABAS resides on a mainframe, NatCDC supports the use of either ADASEL or the ADACDC utility, although NatWorks recommends the use of ADASEL. When ADABAS resides on a UNIX or Windows server, NatCDC supports the use of ADAPLP.

Just so the reader is aware, NatCDC is provided in two versions: NatCDC and NatCDCSP. NatCDC is the historical PLOG processor of NatWorks, and it was designed to handle the PLOG transactions for a single file in a single pass of the PLOG. As the NatCDC product was further developed, the limitation of only processing a single file at a time was recognized, at which time NatWorks developed NatCDCSP. NatCDCSP has the ability to process the PLOG transactions against multiple ADABAS files in a single pass of the PLOG – but can also be used to only handle the transactions from a single file. Both the NatCDC and NatCDCSP modules are included with a NatCDC License.

For further information on NatCDC, please refer to your NatCDC Installation & Operations manual. For general product information please contact NatWorks or your NatWorks representative, or refer to the NatWorks website located at URL http://www.natworks-inc.com.

Integration to Extraction Transformation and Loading (ETL) Tools

To support integration to Data Warehousing products, NatQuery handles the generation of “interface files” or files that describe the physical layout of a NatQuery-generated data extraction process. When these interface files are imported into ETL tools, these tools are enabled to understand the layout of the data being provided by the NatQuery-generated processing, such that most ETL tools can immediately consume and then further process NatQuery-provided extract data.

In current versions of NatQuery, the interface files that can be generated are provided in one of two formats; they can be generated in a DataStage Exchange file (DSX) or a CFD (COBOL File Definition) file.

A DSX file is specific for use with IBM / Ascential Software’s IBM WebSphere Data Integration Suite (formerly known as DataStage) with DSX files being in a “proprietary” structure and format.

A CFD file is a COBOL copybook and is in standard COBOL format.

While the structure and content of these interface files is different, and additionally the data formats between these types of files can be different, both types of interface files accurately describe the physical layout of a data extraction request, and additionally can provide extensive metadata concerning the makeup of the extraction request.

Stand-Alone Data Warehouse Support

As of release 4 of NatQuery, the ability to directly integrate data from ADABAS into databases such as Oracle, SQL Server and MySQL has been introduced. Additionally, NatCDC processing has also been enhanced to support the porting of CDC output directly into these targets.

In many situations where complex data transformations or data cleansing is not needed and the basic requirement is to extract data from ADABAS and load this data into tables in SQL Server, Oracle or MySQL - then NatQuery may be the only tool required.

NatQuery for Processing Data into ADABAS

Beyond the ability to extract data, NatQuery has the ability to generate programs that can process data into ADABAS using sequential files as a data source.

NatQuery generated programs can support such activities as reading Sequential records and:

- **STORE all records into an ADABAS file**
  A Generated process will read all records in a Sequential file and will then STORE these records into the desired target ADABAS file.

- **Match and UPDATE**
  The generated process will look for a record match (based on a defined File Relationship) and will update the record(s) found.

- **Match and UPDATE or STORE**
  The generated process will first look for a record match (based on a defined File Relationship) and if found, an UPDATE against the record(s) will occur; if no matching record is found then a STORE of the record will be performed.

- **Match and DELETE**
  The generated process will look for a match and will then DELETE the matched record(s).

- **Transactions based on a FLAG**
  The generated process will be sensitive to a designated field found on the source sequential file that determines the action to be performed (Match and DELETE, Match and UPDATE, or STORE). Usually, this process is used in conjunction with sequential files that are provided as a result of NatCDC PLOG processing.

The **Process Data into ADABAS** capability provides the ability to process fixed-length sequential data files into an ADABAS file, with these sequential files having been created from a NatQuery-generated extract or some other external source.

Modes of Operation

There are three “modes” of operation that NatQuery will operate under, as follows:

- Administrator Mode
- End-User Mode
- Demo Mode

The mode that NatQuery will operate under is designated by the use (or lack of use) of a **License Key**.

For the purpose of general demonstration, NatQuery can be executed without a **License Key**, in which case NatQuery will operate in **Demo** mode; however Demo mode only offers extremely limited functionality.

When provided with an appropriate key, NatQuery will operate in either **Administrator** mode or **End-User** mode, and depending upon the key used - these may be time-expiring keys.

In most situations, only a single **Administrator** installation of NatQuery is required, although multiple **Administrator** installations may exist. **End-User** installations are designed to enable **End-User** extraction, and in some instances, for example with Data Warehouse initiatives, **End-User** installations may not be required as a single Administrator installation can be used to both configure and then generate all required DWH processing.

Administrator Mode

The Administrator installation is designed for use by a designated “NatQuery Administrator”, and it serves to build, configure and manipulate categories of information that result in an “**Environment Configuration**”. An **Environment Configuration** is a collection of objects and information that relates to a specific Natural server environment, and it is the creation of an **Environment Configuration** that then allows for the intelligent creation of extracts from the Natural server platform using the extraction capabilities of NatQuery.

While an Administrator installation has the capability to create and manipulate an **Environment Configuration**, an **Administrator** installation additionally has the ability to generate requests against the Natural server platform. Therefore, an **Administrator** installation will allow access to all **Administrator** functions of NatQuery as well as allowing access to all functionality found in an **End-User** installation of NatQuery.

End-User Mode

End-User Mode is designed specifically to allow an “End-User” to specify data extraction or audit requests, execute these requests, retrieve the resulting data back to the workstation, and then optionally integrate this data into a wide variety of targets.

By design, an End-User version cannot be used to perform any Administration tasks which allow for the creation of an Environment Configuration.

By referencing the application knowledge captured in an **Environment Configuration** that has been built by an Administrator, a NatQuery End-User need only supply the most basic elements of a data extraction request. To be successful with NatQuery, a user need only provide the following elements of a extraction request:

- The **File**(s) that contain the **Field**(s) to be retrieved,
- The **Field**(s) to be retrieved from the selected **File**(s),
- The **Selection Logic** that specifies what records should be retrieved,
- The definition of any **Variables** (optional),
- The **Target** that the retrieved data should be placed into (Text File, Excel, Access, SQL Server, Oracle, MySQL, etc.).

Further capabilities of the End-User mode can be seen in the section entitled NatQuery for End-User Data Extraction.

Demo Mode

Demo mode is designed only to allow prospective customers / initial users to become acquainted with the look and feel of NatQuery and how it operates. It must be noted that the Demo mode's use is restricted in the following ways:

- **No Environment Configuration Capabilities**
  While all administration functions can be seen, the use of these administrative functions are disabled such that Demo mode cannot be utilized to create or maintain a customer-specific **Environment Configuration**. The Environment Configuration supplied with a Demo Environment Configuration contains the Employees, Vehicles and SAG-TOURS demonstration files provided by Software AG.

- **Limited Generation Capabilities**
  Demo mode is limited to generating Natural data extraction programs against an installation-provided "Demo" **Environment Configuration**.

- **No Integration Capabilities**
  Under Demo mode, NatQuery is disabled from being able to configure or perform either **Automated Movement** or **Automated Execution** against a Natural server platform.

License Keys

As indicated previously, the ability to use NatQuery or NatCDC is based on License Keys.

License Keys are available from NatWorks, Inc or one of NatWorks' designated distributors. Depending upon the requirements / conditions of your NatQuery installation, a License Key may be provided to support NatQuery **Administrator** installation(s) and an additional License Key to optionally support **End-User** installations.

Should you have an issue with the License Keys that were provided to you, an Administrator should contact the distributor or contact NatWorks using the information in the section entitled Contacting NatWorks.

Further information is available on these aspects of License Keys:

- Time Expiring Keys, and
- Switching NatQuery Modes

Time Expiring Keys

Depending upon the requirements / conditions of any given NatQuery installation, License Keys may be provided that are time-expiring. In this case, NatQuery will be fully operational for the appropriate mode (**Administrator** or **End-User**) until the pre-set expiration date. After this date, NatQuery will automatically revert to operating in **Demo** mode.

Switching NatQuery Modes

It is possible to switch the mode under which a given installation of NatQuery is operating by modifying the License Key with a new key that was obtained from NatWorks or through one of its authorized distributors. In this way a time-expired license can be re-activated, or a current **Administrator** installation may be switched to an **End-User** version, or vice-versa.

General Information

NatQuery is written in Visual Basic, and is delivered as a 32-bit application that has been designed to execute on any Microsoft Windows operating system. Such systems would include Windows 98, NT, ME, 2000, XP, Vista, 7, 8 and 8.1; at the present time it is expected that future version of Windows will continue to support Visual Basic applications.

Further General Information is available on the following topics:

- General Workstation Requirements
- General Server Platform Requirements
- Software AG Product Dependencies
- Supported Source Environments
- Security

General Workstation Requirements

NatQuery has the suggested minimum requirements for a workstation installation:

- 486 processor (a Pentium is recommended)
- 6MB Hard Disk (additional space will be required to hold extracted data)
- 2GB RAM

To achieve automatic integration against a remote Natural / ADABAS platform, NatQuery can utilize either File Transfer Protocol (FTP), FTP over SSL (FTPS) or in some cases NatQuery can utilize File Copy operations over a network (such as when the Natural / ADABAS Server resides on a UNIX, LINUX, or Windows machine). In order to achieve such integration the NatQuery workstation must be physically connected / connectable to the server environment over a network, and have appropriate hardware / software installed on both the workstation and the server that will support FTP, FTPS, or automated File Copy operations.

FTP/FTPS Support - Version 5.3.x Forward

From version 5.3.x forward, FTP and FTPS is supported through the use of a licensed third-party product called PowerTCP provided by Dart Communications. The primary reason for this switch was to enable support for FTPS in addition to FTP, both of which PowerTCP fully supports. The PowerTCP components are provided royalty-free to a NatQuery customer, with these components being automatically installed onto the workstation as part of a NatQuery install.

With the introduction of support for FTPS, the use of Microsoft’s WININET.DLL has been jettisoned, such that versions of NatQuery 5.3.0 and higher no longer make any calls to this DLL as was the case with previous versions of NatQuery.

FTP Support Prior to Version 5.3.0

Prior to version 5.3.0 of NatQuery, support for FTP was achieved using calls to a Microsoft-supplied Dynamic Link Library (DLL) called WININET.DLL. However, and as noted above, WININET.DLL provided no support for FTPS.

WININET.DLL is a part of the Microsoft product Internet Explorer, so this product must be installed on the NatQuery workstation. Specific versions of WININET.DLL are required depending upon the host Natural / ADABAS operating system. The following requirements apply:

- For VSE and UNIX servers, WININET.DLL must be version 4.0 or greater.
- For MVS, ZOS and MVS systems, WININET.DLL must be version 5.0 or greater to achieve “Direct FTP” integration. For “In-Direct FTP” integration, then WININET.DLL must be version 4.0 or higher.

Microsoft generally supplies WININET.DLL as part of the Windows operating system, so a NatQuery installation does not supply this DLL. If this DLL is not on your PC, or is installed and a newer version is required, you may obtain this DLL by installing the latest version of Microsoft’s **Internet Explorer**. Internet Explorer is obtainable free-of-charge and is downloadable from Microsoft.

For further information on NatQuery’s use of the Microsoft Dynamic Link Library WININET, including information as to how to determine the version present on a given Windows platform, please refer to the section entitled Microsoft’s WININET.DLL.

General Server Platform Requirements

NatQuery achieves integration to a Natural server platform most commonly through the use of File Transfer Protocol (FTP), FTP over SSL (FTPS), or in some situations the use of File Copy operations across a network.

In all cases, it is expected that Natural will reside on the Server platform, and in most cases ADABAS will reside there as well. Further information on supported versions of Software AG products can be found in the section entitled Software AG Product Dependencies.

Depending upon the level of integration desired between NatQuery residing on a workstation and the remote Natural server, other requirements may exist:

- **Automated Movement Requirements**
  As indicated in previous sections, NatQuery achieves the **Automated Movement** of files and objects between a Windows workstation and the Natural server platform most commonly through the use of Automated FTP / FTPS or in some cases through automated File Copy operations.

  An obvious requirement then is that NatQuery must be physically able to connect to the server platform, and that the server platform will support FTP / FTPS or will alternatively support File Copy operations.

  Depending upon the integration method employed, users of NatQuery must be defined / authorized to utilize the appropriate resources. In the case of FTP / FTPS, NatQuery users will need to be defined to allow the use of FTP / FTPS on the server. For both FTP / FTPS and File Copy, NatQuery users will need to be defined to network security to allow appropriate access permissions.

- **Automated Execution Requirements**
  The Server Requirements to support **Automated Execution** will vary dependent on the server platform itself.

  For mainframe server environments, **Automated Execution** most commonly occurs through the ability to perform FTP / FTPS operations directly against the Job Entry System (JES) on MVS, OS/390 or XA systems, or POWER on VSE, and the FTP / FTPS server needs to be configured to support this. In certain limited situations against mainframe platforms, and when the ability to FTP / FTPS directly into JES or POWER is not allowed - NatWorks may provide a “NatQuery Server” that has the ability to poll for incoming requests and either submits these as they arrive, or at specified intervals.

  For UNIX, LINUX and Windows Natural server platforms, **Automated Execution** can occur either through Remote Execution, or through the use of “polling & submission” software. If Remote Execution will be the method of choice, then the appropriate software such as RSH or REXEC, or third-party products such as PSEXEC (www.syinternals.com) or PLINK (http://www.putty.org) that supports this should be installed and configured. Alternatively, **Automated Execution** can also be achieved by running “poll & submit” software. Poll & Submit software is available from third parties, such as OPALIS (www.opalis.com) or is alternatively available through simple scripts that can be “chroned” or scheduled.

For a list of Specific Server Requirements, please refer to the section entitled Server Checklist.

Software AG Product Dependencies

The following table summarizes the associated Software AG products and their relative dependencies on the server platform.

Product	Requirement	Features/Comments
Natural	Yes	NatQuery will support any version of Natural that is 2.2 or higher. Older versions of Natural may also be supported – contact NatWorks for details. Certain new features of Natural may not be immediately supported, with NatWorks implementing support for these new features at our discretion.
ADABAS	No	NatQuery will support the current SAG supported version of ADABAS, and in most cases will also support the current SAG supported major version of ADABAS minus 1. Older versions of ADABAS may also be supported – contact NatWorks for details. Certain new features of ADABAS may not be immediately supported, with NatWorks implementing support for these new features at our discretion.
PREDICT	No	No specific dependencies
Natural Security	No	No specific dependencies
Natural Interface modules	No	Only required if NatQuery is expected to read from data sources such as DB2 or VSAM.

There are no Software AG product dependencies for NatQuery on the workstation.

Supported Source Environments

The following table summarizes the operational environments that support Natural and are therefore suitable for use with NatQuery.

Platform	**Operating System**
IBM	MVS, Z/OS, XA, VSE, AS/400, I5 series
DEC	OpenVMS
UNIX / Linux	Any platform where Natural is supported
Windows	Any OS above 3.1
Siemens	BS2000, MVS, Z/OS, XA, VSE
Fujitsu	MVS, Z/OS, XA, VSE
Hitachi	MVS, Z/OS, XA, VSE

Security

If security is an issue at a customer site, then this is most likely already being addressed by using one or more security packages specific to Natural / ADABAS software.

For example, in order to have a user be able to automatically move an execution request onto the server platform, either FTP / FTPS or Network Copy is likely to be used, and these integration mechanisms can be configured to allow use only after a valid User-ID and Password is provided by the user.

To assist in guarding against attacks on NatQuery configuration files, encryption is employed to detect when unauthorized changes are made against workstation Environment Configuration files, such that only users who are authorized Administrators can manipulate the Environment Configuration successfully.

NatQuery natively embraces other aspects of security within a Natural / ADABAS environment, specifically:

- **Natural Security**
  In order to conform to **Natural Security**, NatQuery can be instructed to solicit both a User-ID and Natural Security password that then becomes imbedded into submitted batch processes just prior to the actual submission of the processes. This option is enabled through the **Natural Server Information** window. If Natural Security support is enabled, NatQuery will prompt for a Natural Security User-ID and password each time an extract request is submitted to the server.

- **ADABAS Security**
  NatQuery provides support for ADABAS file-level security by generating the necessary Natural statements for each file access that will require a password. This option is set for each individual file. If ADABAS security is turned on for a given file, NatQuery will prompt for a password for that file each time a query containing fields from that file is submitted to the server.

General Note:
While NatQuery can be configured to prompt for appropriate FTP / FTPS or Network User ID and Passwords, it should be noted that NatQuery does not store any passwords beyond the duration of the current session. Password values will be retained in memory for re-use throughout the duration of a given user’s NatQuery session only.

Pre-Install Checklist

The purpose of this section is to provide an overview of both the information that must be available and the actions that should take place before an installation and configuration of NatQuery is performed. While an installation of NatQuery can occur in minutes, the process of configuring NatQuery is more involved.

This section presents, in a checklist format, the pre-requisites and information that should be gathered and known in advance before a configuration is started so that configuration can be easily and rapidly accomplished.

The areas covered by this checklist are:

- Workstation Checklist, and
- Server Checklist

Workstation Checklist

NatQuery is an application that is installed onto a Microsoft Windows workstation. The requirements of this workstation are as follows:

□ **Microsoft Windows Operating System (95, 98, NT, 2000, ME, XP, Vista, 7 and 8)**
While not being an absolute requirement, it is advised that the Windows Operating System be configured to use English as its regional language setting.

□ **6 MB of Free Hard Disk**
To accomplish an installation, approximately 6 MB of free space should be available on the workstation's hard drive. Additional free hard disk space should be available to hold extracted data.

□ **FTP Connectivity to the Remote Natural / ADABAS platform**
In order to achieve automated integration between a NatQuery workstation and the remote Natural / ADABAS server platform, NatQuery can utilize either FTP / FTPS (client software support for FTP and FTPS is installed with NatQuery) or in some cases may use Network File Copy operations. If there is no intention of achieving automated integration, then FTP / FTPS or Network File Copy operations is not a requirement.

□ **Microsoft Internet Explorer (Optional as of NatQuery version 5.3.0 and above)**
As of version 5.3.0 of NatQuery, there is no longer a requirement that WININET.DLL be available as it is no longer used in these versions.

In order to achieve automated integration between a NatQuery workstation and the remote Natural / ADABAS server platform, NatQuery utilizes FTP / FTPS over TCP/IP or can alternatively use automated File Copy operations.

For versions of NatQuery 5.3.x or higher, FTP and FTPS is supported through the use of a third party component that is provided royalty-free with an installation of NatQuery. If the version of NatQuery then is 5.3.x or higher – then there is no requirement for Internet Explorer.

For versions of NatQuery prior to 5.3.0, the implementation of FTP by NatQuery utilized a Dynamic Link Library (DLL) module called WININET.DLL that is made available by an installation of Microsoft Internet Explorer. As of this writing, WININET.DLL does not provide support for FTPS.

If the remote Natural / ADABAS system is MVS, Z/OS or XA and it is intended that FTP operations be performed directly into the Job Entry System (JES), then the version of WININET.DLL must be version 5 or higher.

For all other situations where automated FTP integration is desired, the version of

WININET.DLL should be 4.0 or higher.

If Microsoft’s Internet Explorer product is not installed on the workstation, then either manual integration will be used or Internet Explorer should be installed. Internet Explorer is obtainable free-of-charge and is downloadable from the Microsoft website.

For further information on NatQuery’s use of Microsoft’s WININET.DLL, including information as to how to determine the version present on a given Windows platform, please refer to the section in the NatQuery Installation and Operations Guide entitled Microsoft’s WININET.DLL.

Server Checklist

The following requirements exist for the Natural / ADABAS server environment.

□ **Software AG’s Natural**
It is a requirement that Software AG’s Natural 4GL must reside on the server, and it is strongly suggested that the version of Natural be currently under support by Software AG.

□ **Software AG’s ADABAS**
While not being an absolute requirement (due to the fact that Natural can be used against data stores other than ADABAS), it is a general requirement that ADABAS should exist on the server.

□ **Batch Natural Nucleus**
As all NatQuery-generated processing occurs in batch, batch support for Natural is critical. In some cases, it may be advisable to create a NATPARM module that is specific for use with NatQuery-generated batch processing.

□ **Available Disk Space**
Sufficient disk space on the server should be available to contain NatQuery-related processing files and extracted data.

□ **Disk Files Naming Conventions**
You should be familiar with the naming convention in use for files on the Natural / ADABAS server.

For MVS, Z/OS, XA systems, NatWorks suggests that server-side datasets required for NatQuery will be named with a high-level qualifier that equates to the User-ID of a given NatQuery user.

For VSE systems, NatWorks suggests the use of a VSAM Catalog to contain required NatQuery files; VSAM is recommended due to its ability to allow a file to easily expand beyond initially allocated file extents.

For UNIX and Windows Natural server platforms, a directory structure should be created to support NatQuery processing.

□ **Server-Based Security**
The users of NatQuery must be authorized to access the Natural / ADABAS server platform in batch. This may require attention to server security packages, and may also require attention to the FTP / FTPS Service (if applicable).

For MVS, Z/OS and XA systems, security should enable a user to allocate and access datasets that are defined with the user's User-ID as the high-level qualifier.

For VSE systems, security should enable a user to access the required VSAM catalog.

For UNIX and Windows Natural server platforms, the user must be granted Read / Write access to the appropriate Server directories.

□ **User-IDs**
The Administrator should have available a list of all other users who will be initially using NatQuery to generate extracts, along with each of these user's server User-Ids.

□ **Access to the SYSTRANS/SYSOBJH Utility**
Normal NatQuery operations occasionally depend on Natural modules that have been built by NatWorks and which get installed into the Natural Server environment. These modules are provided in both SYSTRANS and SYSOBJH format for loading into the Natural Server environment, such that the Administrator must have access to SYSTRANS and or SYSOBJH - with SYSTRANS being the older utility/methodology for handling Natural objects and SYSOBJH being the newer utility/methodology.

□ **Machine Resources & Processing Time**
NatQuery is a tool that is "dual use": It can be configured to be used as an End-User query tool, it can be configured to be used as a Data Warehouse extraction tool, or it can be configured to provide both capabilities.

In its capacity to serve as an End-User Query tool, one of NatQuery's features is that it has the ability to determine the optimal access path (Descriptor / Sub-Descriptor / Hyper-Descriptor / Super-Descriptor / ISN) to use to resolve a given extract based on user-entered data-selection criteria. In order to accomplish this, NatQuery utilizes "**Descriptor Statistics**", with these "**Descriptor Statistics**" being created as a result of executing NatQuery-generated Natural programs that capture the required information.

If NatQuery is going to be used as an End-User query tool, then processing time will be required on the Natural / ADABAS server machine so that the required "**Descriptor Statistic**" capture program(s) can execute. The time required to execute a given NatQuery-generated Descriptor Statistic capture program will vary from file to file, and will be dependent upon the number of records in the file and the number of Descriptors and Super Descriptors present in a given file.

In situations where NatQuery is to be used solely for Data Warehousing, then accurate "**Descriptor Statistics**" become non-critical due to the fact that in most DWH situations the required data extraction programs will need to read through entire files (I.E. the Natural READ PHYSICAL I/O statement will be utilized – or the ADAULD / ADACMP utilities will be used). In this situation, having accurate Descriptor Statistics is not required, and NatQuery provides the capability to bypass their capture: However

NatQuery will still need to be told how many records physically exist in each file.

□ **Server Disk Files**
The naming convention in use for files on the Natural / ADABAS server should be known.

For MVS systems, NatWorks suggests that disk files associated with individual users be defined so that they have a high-level qualifier that equates to the individual user's User-ID.

For VSE systems, NatWorks suggests that a VSAM Catalog be created to contain all server-side files, and within this catalog disk files associated with individual users be defined so that they have a high-level qualifier that equates to the individual user's User-ID.

For UNIX and Windows Natural server platforms, NatQuery provides a default naming convention that can be easily changed as need requires.

□ **Server Disk Attributes**
Specific to situations where the Natural server resides on a mainframe platform, NatQuery will generate Job Control Language (JCL) streams that are designed to execute in batch. In order for these generated streams to execute as expected, NatQuery must generate JCL to allocate disk files to receive extracted data. In order to accurately calculate disk estimates, NatQuery needs to understand the general attributes for the disk drives that will be used to hold extracted data. Specifically, you will need to know:

- **Device Blocksize** _________
  The NatQuery default value for this will be 28332.

- **Blocks per Track** _________
  The NatQuery default value for this will be 2.

- **Tracks per Cylinder** _________
  The NatQuery default value for this will be 15.

- **Cylinders per Volume** _________
  The NatQuery default value for this will be 2226.

FTP / FTPS Requirements

In order to achieve automated FTP / FTPS integration between NatQuery on a workstation and the Natural / ADABAS server, FTP / FTPS connectivity must be available on this server. Without FTP / FTPS connectivity, integration can only be accomplished manually or possibly with manual or automated File Copy operations.

If FTP / FTPS connectivity is desired, then the following information is required for NatQuery configuration:

□ **Server IP Address**
The IP address or URL of the Natural / ADABAS server environment should be known.

□ **Proxy Server IP Address**
The IP address or name (URL) of any proxy server being used to communicate with the mainframe at your site should be known. In most cases, proxy servers are not used, and their use may prove problematic for FTP / FTPS.

□ **FTP / FTPS Accessibility to Internal Reader Queues (Mainframe servers only)**
On both MVS (MVS, OS/390, XA) systems and VSE systems, FTP / FTPS can be configured to allow FTP **PUT** operations initiated from a workstation to place files (JCL) directly into the remote Natural / ADABAS system's internal reader.

Prior to configuring NatQuery, it should be ascertained whether or not FTP / FTPS operations are already configured (or this configuration can be allowed) so that **PUT** operations from a workstation can place this JCL into the Natural / ADABAS server platform's internal reader. For MVS systems this would be the Job Entry System (JES), for VSE systems this would be POWER.

If FTP **PUT** operations are allowed or can be allowed against the server's internal reader queue, then the NatQuery "**Direct-FTP**" method can be utilized which will greatly simplify the configuration process.

If FTP **PUT** operations are NOT allowed against the server's internal reader, then "Just FTP" or more manual methods of integration can be utilized.

Remote Execution Requirements

When NatQuery will be used against UNIX (and variants) or a Windows Natural server platform, then automated remote execution can either be accomplished through the use of Remote Execution utilities (for example: SSH, RSH or REXEC against UNIX or third-party utilities such as PSEXEC or PLINK).

If the use of Remote Execution utilities represent too much of a security risk, then automated execution can still be achieved through the use of server side “Poll & Submit” software. This type of software can be purchased from third parties, or can be easily constructed using in-house resources. In most cases, NatWorks can also provide examples of “poll and submit” scripts / processes.

Natural Requirements

The following are considerations and requirements for the Natural environment.

□ **Natural Environment Values**
Various aspects of any given Natural Environment are parameterized to accommodate internationalization and operational flexibility. The following values should be known:

- **Decimal Character** __________
  This would be the value of the Natural Parameter DC (Decimal Character); the NatQuery default for this is "**.**" (Period).

- **Variable Prefix Character** __________
  This is a character that is typically used to prefix "local variables" in a Natural program; the NatQuery default for this value is "**#**" (Pound Sign).

- **Natural Command Delimiter** __________
  This would be the value of the Natural Parameter ID (Input Delimiter); the NatQuery default for this is "**,**" (Comma).

□ **Data Definition Modules (DDMs)**
DDMs should be available on the server for the source files intended for extraction. In situations where a Trial or Proof of Concept (POC) is to be conducted, it is highly desirable that DDMs be available on the workstation environment prior to even beginning the Trial or Proof of Concept, as this will speed the initial configuration.

The following considerations apply to these DDMs:

- **Sub-Descriptors and Super-Descriptor Component Definitions**
  If End-User extraction is to be supported, then it is a requirement that DDMs be generated so that the components of Sub-Descriptors and Super-Descriptors are described in each DDM that will be made available to NatQuery.

  To enable the generation of this component information from Predict, DDMs should be generated so that the generation option **General Comments** is set to "**Y**".

- **Occurrence Information**
  It is strongly recommended that DDMs be generated so that any documented occurrence information in Predict (if present) will be in the DDM under the **Remarks** column.

  If this information is not documented in Predict, or Predict is not present, then the maximum number of occurrences for each Multi-Value Field (MU) or Periodic-Group (PE) may be manually entered or can be programmatically determined through the generation of NatQuery Data Discovery programs.

To enable the generation of any documented occurrence information from Predict into a DDM, the generation option **Line Comments** should be set to "**O**" (the letter "O"), when a DDM is generated from Predict.

- **DDMs Must Exactly Match FDTs (NatCDC requirement only)**
  When NatQuery will be used with NatCDC it is a requirement that DDMs used for PLOG processing ***MUST*** exactly match the corresponding File Definition Table (FDT) for the same file.

  To assist in this requirement, NatQuery has the capability to create a DDM from a FDT, as well as being able to automatically download both a DDM and an FDT, and then merge the DDM with the FDT.

- **Negative Numeric Fields**
  DDMs that contain numeric fields should be identified. Unless otherwise instructed, NatQuery will by default assume that the contents of numeric fields are all positive values and will therefore not attempt to provide a leading sign byte when these elements are extracted.

  This processing is by design, and is intended to save disk space that would normally be used to hold the sign byte for each field which in most cases will always be a positive value anyway.

  NatQuery can however be instructed to handle sign bytes for designated fields in specific files, so that a sign byte can always be automatically output whenever the field is selected for output.

- **File Relationships**
  If the intent of using NatQuery is to produce multi-file extracts or extracts with automatic "File Lookups", then NatQuery will need to be provided with information on the "File Relationships" that will then support this automatic file linking or "joins". This means that the Administrator must have an understanding of the Descriptors or Super-Descriptors that support the underlying "File Relationships"; they must have an understanding of the cardinality of these relationships (i.e., 1:1, 1:many, etc.), and must further know how these Descriptors or Super-Descriptors are properly initialized in order to link related records.

□ **NatQuery Natural Library Name**
A Natural Library should be designated from which all NatQuery generated programs will be run. While a Natural Library must be named, NatQuery-generated programs can be run as "adhoc" programs (which are neither saved or cataloged, just run), or these programs can be saved and then run, or catalogued and then executed, or saved and

catalogued and then run or Executed.

□ **Batch Natural JCL / Script Example**
The physical execution of NatQuery processes occur in a batch environment on the Natural server platform. In order to support the automation of user-generated processes, NatQuery generates the execution JCL / Script for a given request and it does this using a template-based approach.

To facilitate the configuration of NatQuery, a working example of a batch Natural JCL / Script is an exceedingly helpful reference point. By referencing a working JCL or Script that properly executes in the intended Natural environment, the process of creating the required NatQuery templates is hugely simplified.

When approaching a Proof of Concept or a Trial of NatQuery, it is highly recommended that a prospective customer provide a copy of a working JCL/Script Stream that successfully executes Natural against the suggested target environment to either NatWorks or the designated NatWorks distributor **PRIOR** to beginning the trial or POC, as either NatWorks or the designated NatWorks Distributor can take that JCL/Script and use it to create the basic JCL/Script templates that will be required for success.

As the single most time-consuming effort in a NatQuery installation is getting JCL/Script templates correct - much time can be saved by allowing NatWorks to bring their expertise to bear on this aspect of configuration.

Natural Security Checklist

The following are considerations and requirements when the Software AG product Natural Security is utilized.

□ **Access to designated NatQuery Natural Library**
All Administrators and users of NatQuery should be defined through Natural Security to access the designated NatQuery Natural Library.

□ **Access to SYSTRANS / SYSOBJH Utility**
In order to facilitate the introduction of DDMs / FDTs into NatQuery, NatQuery can utilize the SYSTRANS or the SYSOBJH utility of Natural, executed through batch. If it is desired to use NatQuery's capability to process SYSTRANS / SYSOBJH requests, then at least one Administrator must be given access to the SYSTRANS / SYSOBJH utility.

□ **Administrator Authorized To Have Read Access to All Desired Source Files**
In situations where NatQuery is to be used as an End-User query tool; it is strongly recommended that "Descriptor Statistics" capture programs be run that will capture the required information. In order to accomplish this, at least one NatQuery Administrator must be given Read Access to all desired source files.

□ **Users Authorized To Have Read Access to Allowed Source Files**
In order to allow an End-User to retrieve data from a given ADABAS source file, each user (either individually or as a group) must be given access to the desired ADABAS source files.

Installing NatQuery

This section details the general procedure for installing NatQuery so that it can be subsequently configured to operate against a Natural / ADABAS environment that resides on MVS (or variant), VSE, UNIX (and variants), OpenVMS or Windows.

The process of installing NatQuery will generally include the following two steps:

- Obtain NatQuery Installation Software and License Keys, and
- Install NatQuery

Obtain NatQuery Installation Software and License Keys

In order to install NatQuery the installation software must be obtained. This software may be made available to you through a designated partner of NatWorks, it may be provided on a NatWorks CD /DVD, it may be made available for download from the Downloads section of the NatWorks website located at www.natworks-inc.com, or it may be provided by a designated partner's website.

If the software will be obtained from the NatWorks website, this access will require a User-ID and Password. The User-ID and Password will be provided to you by your vendor, or directly by NatWorks, Inc.

In order to fully utilize NatQuery subsequent to installation, you will need one or more License Keys that will be provided to the NatQuery configuration process.

If you have not been given a User-ID and Password to access the NatWorks website, or have not been given a License Key that will activate NatQuery / NatCDC – please secure these before continuing.

Install NatQuery

If the NatQuery installation software is provided on CD / DVD, then in most cases the act of putting the CD / DVD into the drive bay and closing the tray will be sufficient to launch the install process.

If the NatQuery software is downloaded from the NatWorks website or the CD / DVD does not "auto run", then a file will be provided that is named:

Install_NatQuery.exe

To install NatQuery, the above file must be executed. This is most easily accomplished by locating the above named file using Windows Explorer, and once located, double right-clicking the file. This action will begin the installation process of NatQuery.

The installation process will react differently depending on whether NatQuery has been previously installed on the workstation or not.

If NatQuery has not previously been installed, then the user should refer to the section entitled

Installing NatQuery When Not Currently Installed immediately following this section.

If NatQuery was previously installed, then the user should refer to the section entitled

Installing NatQuery When Previously Installed further below.

Installing NatQuery When Not Currently Installed

During a new installation process, the user will be presented with several prompts. What follows are the windows that will generally be presented with a new installation after the file Install_NatQuery.exe has been executed:

1. **Open File - Security Warning**
   Immediately after executing the Install_NatQuery.exe file, the Windows may respond with a security warning concerning the inability to verify who created the installation file. In response to such a prompt, the user should indicate that they want to Run the installation file.

2. **Welcome to the NatQuery Installation Wizard**
   This prompt welcomes the user to the installation wizard. Click the **Next** button to proceed.

3. **NatQuery Setup / User Information**
   This window prompts for information specific to the user of NatQuery. It is suggested that the user enter their Full Name (required), as well as the organization they work for (optional).

   The User Information window also asks if the installation of NatQuery will be shared by any other user of the installation machine, or whether it should be specific to a single user. The suggested response is to accept the default selection (I.E. "**Anyone who uses this computer**"), and then click the **Next** button.

4. **NatQuery Setup / Destination Folder**
   This window prompts the user for the path of the folder that NatQuery will be installed into, with the installation process providing a default directory.

   For 32-bit machines this Destination Folder will usually be:

   C:\Program Files\NatQuery\

   On 64-bit machines, this Destination Folder will usually be:

   C:\Program Files (x86)\NatQuery\

   It is strongly suggested that the user accept the provided default and then continue the installation process by clicking the **Next** button.

5. **NatQuery Setup / Ready to Install the Application**
   This window prompts the user for confirmation to begin the installation process. Click the **Next** button to begin the installation process.

   The window's title will then change to be "NatQuery Setup", and will then show a progress bar showing its status.

6. **NatQuery has been Successfully Installed**
   This window informs the user that the installation has been completed successfully. In response the user will typically click the **Finish** button to complete the installation process.

   The user may now proceed to the section entitled Starting NatQuery for the First Time.

IMPORTANT NOTE - USE OF USER PROFILE DIRECTORY

In versions of NatQuery prior to 5.4.0, an installation of NatQuery would create a directory called NatQuery in the "Program Files" or "Program Files (x86)\" directory. After installation, NatQuery would then, by default, write information back to that installation path or a sub-directory under that path. These files included such things as the License Key file ("license.key") and user-specific configuration file ("config.cfg"), among other key files. In this way all files that pertained to NatQuery could all be kept together under one directory structure.

With the advent of Windows 7 and above, Microsoft opted to better secure the Program Files / Program Files (x86) directories, and they therefore changed the default permissions of the installation created directories and sub-directories to be read-only.

This issue then caused Permission errors when NatQuery attempted to write an essential file in the NatQuery installation path under Windows 7 and 8 operating systems.

Prior to the release of NatQuery 5.4.0, the only solution to this issue was to change the permissions for the NatQuery installation folder to allow the user to have Read, Write and Erase / Delete permissions for the NatQuery installation directory and it's sub-directories.

With the release of NatQuery 5.4.0, NatWorks changed where NatQuery would write files by default, and this change embraced the concept of using a user's default User Profile Directory, which is the same directory into which a user's "My Documents" directory is created.

When NatQuery is installed, it is installed into the default Program Files directory, which on newer operating systems is typically "c:\Program Files (x86)\". When NatQuery is first executed by a given user, NatQuery will check for the existence of a "NatQuery" directory in the User's Profile Directory, and if this is not found NatQuery will automatically create the directory and will then copy into this directory various directories and files from the install directory.

From initial execution on, when NatQuery writes and new file(s), they will be created, depending on the action occurring, in either the designated "Environment Configuration Path", the designated "Output Path", the designated "Query Path" and/or the NatQuery directory found in the User's Profile Directory.

This solution should be transparent to the end-user or Administrator, with the only issue being that when an uninstall of NatQuery is performed, the uninstall processing will not remove or even look at any of the files created by NatQuery which are found in the NatQuery directory under the User's Profile Directory.

Installing NatQuery When Previously Installed

If NatQuery was installed previously, the following prompts will generally be seen.

1. **Uninstall Existing Version**
   In situations where a previous version of NatQuery was installed, the installation process will typically present a message box asking if the previous version should be uninstalled.

   It is NatWorks suggestion that the user respond with **YES** so that the install process **WILL** begin with the uninstall of the previous version.

   Uninstalling a currently operational version will have NO EFFECT on any configured information that may have been built previously – only the files directly related to the currently installed version will be removed.

2. **Welcome to NatQuery Installation Wizard**
   This prompt welcomes the user to the installation wizard. Click the **Next** button to proceed.

3. **License Agreement a.k.a. EULA (End User License Agreement)**
   This window prompts the user to review the NatQuery License Agreement. In order to proceed with the installation, click the radio button that indicates acceptance of the NatQuery License Agreement, then click the **Next** button. The EULA can then be reviewed at the user's convenience from the NatQuery desk top by clicking into Help.

4. **User Information**
   This window prompts for information specific to the user of NatQuery. It is suggested that the user enter their name, as well as the organization they work for.

   The **User Information** window also asks if the installation of NatQuery will be shared by any user of the installation machine, or whether it should be specific to a single user. The suggested response is to accept the default selection (I.E. "Anyone who uses this computer"), and then click the **Next** button.

5. **Destination Folder**
   This window prompts the user for the path of the folder that NatQuery will be installed into, with the installation process providing a default directory.

   For 32-bit machines this Destination Folder will usually be:

   C:\Program Files\NatQuery\

   On 64-bit machines, this Destination Folder will usually be:

   C:\Program Files (x86)\NatQuery\

   It is strongly suggested that the user accept the provided default and then continue the installation process by clicking the **Next** button.

6. **Ready to install the Application**
   This window prompts the user for confirmation to begin the installation process. To allow the installation to begin, click the **Next** button.

7. **NatQuery has been successfully installed**
   This window informs the user that the installation has been completed successfully.

   The user may now proceed to the section entitled Starting NatQuery for the First Time.

Uninstalling NatQuery

To uninstall NatQuery, perform the following (the following instructions are for Windows 7):

1. Click the **Start** button on the Windows Task bar.

2. Click on the **Control Panel** icon.

3. On the **Control Panel** window, click on the **Programs and Features** icon.

4. On the **Uninstall or change a program** window, locate the NatQuery application in the scrollable list box and then highlight NatQuery by single left-clicking on it.

5. With the NatQuery application highlighted, click the **Uninstall** option.

The above steps will uninstall the NatQuery application from the workstation, however any and all currently-existing **Environment Configuration** information as well as any information (such as Query Specifications) or data created by the NatQuery application or extract processes will be left intact.

If your intent is to completely remove NatQuery, this can easily be accomplish subsequent to the uninstall process described above by using the Windows Explorer application to delete the NatQuery installation directory, and to additionally remove the NatQuery directory found under the user's profile directory (usually found under **c:\Users\*username*\** (where "username" is the user's user name); these actions will effectively remove all traces of NatQuery from the given workstation.

To remove all traces of NatQuery from the server platform, it is usually sufficient to delete the contents of the designated NatQuery Natural target library, and to then also delete any server-side output files that may have been created via NatQuery generated processes.

Starting NatQuery

A successful installation of NatQuery should create a desktop icon for NatQuery, and should additionally create a menu item under the **All Programs** link available from the **Start** button's menu. To start NatQuery, the user may either double-click the desktop icon, or may locate the NatQuery icon under Start / All Programs and then single-click the NatQuery icon. Either of these actions will start NatQuery.

Starting NatQuery for the First Time

When NatQuery is started for the first time and no valid License Key information exists, NatQuery will provide a **License Key Notice** message box, with this notice informing the user that NatQuery does not currently have a valid License Key. Click **OK** to this prompt.

NatQuery will then display an animated NatQuery “splash” screen, which will subsequently be shown each time NatQuery is started. The splash screen will indicate the version number for the current installation of NatQuery as well as the mode: Administrative, End-User, or Demo. Initially, the mode will show as Demo until NatQuery is given a valid License Key(s).

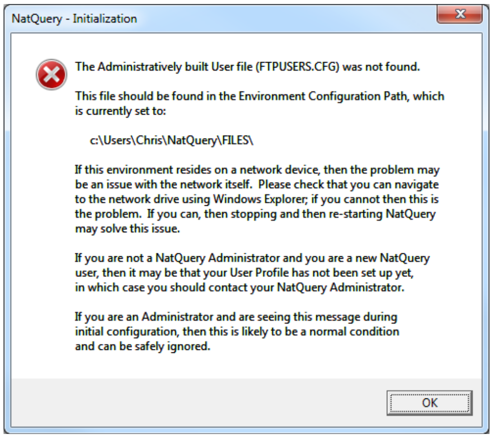
After the splash screen disappears, and in situations where NatQuery has just been installed for the first time, NatQuery will typically display a **NatQuery Configuration** window that is specifically designed to allow entry of NatQuery **License Key(s)**.

The **License Keys** tab of the **NatQuery Configuration** function allows for the entry or modification of NatQuery and / or NatCDC License Keys. This window will be automatically displayed with the first execution of NatQuery, or can be alternatively invoked at a later time by clicking the **Administer** drop-down menu, then clicking the **NatQuery Configuration** function, then clicking on the **License Keys** tab.

On the **License Keys** tab, enter the Administrator License Key provided to you for NatQuery. If appropriate to your situation, the License Key for NatCDC may be entered as well; if NatCDC is not to be installed, then leave the NatCDC License key fields blank.

When the appropriate License Key(s) have been added, the user may now click the **Ok** button on the **NatQuery Configuration** window.

In situations where NatQuery is being installed for the first time or if no Environment Configuration Information was provided to the previous installation, NatQuery will now produce a message box similar to the following:



The Administratively built User file (FTPUSERS.CFG) was not found.

This file should be found in the Environment Configuration Path, which is currently set to:

c:\Users\Chris\NatQuery\FILES\

If this environment resides on a network device, then the problem may be an issue with the network itself. Please check that you can navigate to the network drive using Windows Explorer; if you cannot then this is the problem. If you can, then stopping and then re-starting NatQuery may solve this issue.

If you are not a NatQuery Administrator and you are a new NatQuery user, then it may be that your User Profile has not been set up yet, in which case you should contact your NatQuery Administrator.

If you are an Administrator and are seeing this message during initial configuration, then this is likely to be a normal condition and can be safely ignored.

Figure 1 – User File Not Found

This error is normal, and simply indicates that NatQuery has not yet been provided with configuration information that defines who the user(s) of NatQuery are. Click **Ok** to this prompt.

The user should now be placed on the NatQuery desktop, with all toolbar icons disabled. At this point, the user may proceed with NatQuery Configuration.

Note 1:
The installation and configuration of NatCDC is described in a separate manual. If you do not intend to use NatCDC, or you are currently only interested in configuring NatQuery, then the NatCDC License Key should be left blank and can then be entered at a later time.

Note 2:
In order to operate in any mode other than DEMO mode, NatQuery ***must*** be given a valid License Key. If DEMO mode is desired, then the rest of this document becomes irrelevant, as DEMO mode will specifically block Administrative functions that are designed to allow NatQuery to be configured to a customer's environment.

Overview of NatQuery Configuration

Once NatQuery is installed, the process of configuring NatQuery can begin through the use of an Administrator version of NatQuery (an installation of NatQuery that has been given an Administrator License Key). Configuring NatQuery involves the creation and capture of information that allows NatQuery to integrate with Natural / ADABAS on either the same or a different platform.

The intention of this section is to provide the reader with a general overview of the Configuration Process and the Information required. If the reader would prefer to skip this and move forward with the configuration, then the user is encouraged to jump to the section entitled Creating a NatQuery Configuration.

Generally speaking, there are two categories of information that comprise a complete NatQuery Configuration:

- Connectivity Information**,** and
- Environment Information

Connectivity Information

Connectivity Information provides NatQuery with information on how integration will be achieved between a NatQuery workstation and a remote Natural server platform.

Connectivity Information will therefore involve defining the:

- **Method of Integration**
  NatQuery supports the ability to interact with the remote Natural server platform using either FTP, FTPS or File Copy operations performed over a shared network. NatQuery must be provided with configuration information that supports the intended integration so that it can properly interact with its server in a specific customer environment.

  The **Method of Integration**, described in greater below, indicates how (or if) the automated movement of generated objects between the NatQuery workstation and the Natural / ADABAS server platform will occur, and how (or if) automated execution of NatQuery-generated processes will occur.

- **NatQuery Users**
  In order for a user to interact with the remote Natural server, that user must be defined to NatQuery. One of the primary functions of defining a user is to link a user to the file names that the user will utilize to interact with the Natural server platform, as well as to define the permissions of that user.

- **JCL / Scripts**
  The processes that NatQuery generates are designed to execute in a batch environment, and there are several unique types of processes that NatQuery may invoke. To support each of these unique processes, as well as other processes that the user may create, NatQuery provides JCL / Script templates. These templates will require customization so that they will successfully operate as intended in a specific customer environment.

To handle the automated movement of files between a NatQuery installation and the Natural server platform, two methods are supported:

- **Automated FTP / FTPS** operations or,
- **Automated File Copy** operations made across a network.

NatQuery is primarily designed to support **Automated FTP / FTPS** against all Natural server platforms that support and provide an FTP / FTPS connection from the workstation environment.

NatQuery can support **Automated File Copy** operations against a Natural server on a Windows, UNIX or Linux platform (if shared disk is available).

As a last resort, it should be noted that NatQuery does not absolutely require any automated movement between the NatQuery workstation and the Natural server platform to be able to generate required processing. In such cases, manual intervention will be required. This manual intervention will handle the manual movement of NatQuery-generated processes onto the Natural server platform (for the execution of required processing), the manual execution of the JCL / Script, and the movement of other installation-provided files onto the server that will then help to support NatQuery processing.

On the retrieval side of the extraction process, there will be the need for manual movement of data delivered by generated processes to the NatQuery workstation or elsewhere.

In regards to what objects / files are physically moved with automated movement to support an execution request; this will vary depending on the Natural server platform.

For **MVS** and **VSE** Natural server platforms, a NatQuery request will typically be comprised of a generated Natural program being moved into the mainframe environment by being imbedded into a generated JCL stream that executes the generated program.

For **UNIX, OpenVMS** and **Windows** Natural server platforms, a single NatQuery request will eventually be comprised of 4 separate text files, handled initially as a single file with designated separators. These files would be:

- The generated Natural program.

- The Script that will execute the process.

- A CMSYNIN input file (Natural commands as well as data to be read by INPUT statements) that will be used by the Script.

- A CMOBJIN file (data that will only be read by INPUT statements) that will be used by the Script.

The generation of required JCL / Scripts is based on administratively provided templates which are built based on the use of NatQuery installation-provided examples that are customized to specific site requirements during the configuration process.

The intent of automated movement is to have NatQuery place onto the server everything that is needed to provide for the execution of a request, such that the only other consideration is how the request will be executed.

The options for how automated execution of NatQuery-generated processes will occur are dependent on the method of automated movement, as well as the Natural server platform itself.

For situations where **MVS** or **VSE** is the Natural server platform, the use of FTP / FTPS to accomplish automated movement can also supply a solution for automated execution in that FTP / FTPS operations against either of these two mainframe environments support the ability to directly FTP / FTPS files into the internal reader. Taking advantage of this capability means that NatQuery's automated FTP / FTPS operations can support direct FTP / FTPS into the JES (**MVS** systems) or POWER (**VSE** systems); thus accomplishing automated execution.

For situations where **UNIX** (and variants), **OpenVMS**, or **Windows** is the Natural server platform, then either the use of Remote Execution / Remote Shell utilities (invoked automatically via command line by NatQuery), or the use of server based processes that “poll and submit” can accomplish automated execution. Such processing may be made available by third-party vendors, home grown solutions, or such solutions may also be provided by NatWorks.

As an execution method of last resort, it should be noted that there is no absolute requirement for automated execution against the server platform to allow NatQuery to generate required processing and move these processes onto the server. In such situations, execution of generated processes will be occurring under some other means such as manual intervention or external automation.

Environment Information

Environment information pertains to information that is directly related to the source ADABAS database(s) and files. The following information is required to build an **Environment Configuration**:

- **Data Definition Modules (DDMs)**
  DDMs are a basic building block for Natural access to ADABAS, and DDMs are used by NatQuery to gain basic information on specific ADABAS files or views of these files.

  DDMs can be obtained automatically from the ADABAS platform through automated movement and automated execution of NatQuery-generated processing (utilizing SYSTRANS or SYSOBJH), or DDMs can be imported into NatQuery (utilizing several methods such as the use of SYSTRANS output, SYSOBJH output, or an Entire Connection output that is manually moved onto the NatQuery workstation).

- **File Relationships Information**
  File Relationship Information allows NatQuery to provide automatic linking between DDMs (I.E. automatic "joins").

  File Relationship Information is optional, and for Data Warehousing extractions where only single-file extracts will be the norm, this information is typically not needed. For End-User extractions however, providing File Relationship Information allows users to automatically link / join multiple ADABAS files together into a single logical data extraction request, without requiring the user to have any knowledge of how the link is physically implemented.

- **Descriptor Statistics**
  Descriptor Statistic information describes all of the access paths (keys / indices) available with a given DDM, and allows for the capture of information about the characteristics of these access paths or keys / indices. This information can either be automatically captured by a NatQuery-generated process, or may be manually supplied. With this information available, NatQuery gains a level of intelligence that allows for a automated determination of the optimal access path to resolve a given query, based on user-supplied Selection Logic.

- **Occurrence Information**
  Occurrence Information describes the default occurrence specifications for each recurring field, as well as specifying the maximum occurrence specification. This information then controls how a given data extract request will reference the recurring field structures that may exist in ADABAS such as Multi-Valued fields (MUs), Periodic-Groups (PEs) or MUs in PEs when these fields are referenced by a user for extraction.

- **Sign Byte Information**
  Sign Byte Information provides NatQuery with information on how to handle the extracted output of numeric fields. By default, NatQuery will output numeric fields with no sign byte, essentially assuming that all numeric-type fields are positive (and thereby saving space in the extract file). If a sign byte is required for any source field, then providing Sign Byte Information for this field and other fields that require this will instruct NatQuery to output the designated numeric field(s) with a leading sign byte position / character in the final output.

- **I/O Parameters**
  I/O Parameters generally describes program generation rules that are used when NatQuery determines what the best I/O statement is to resolve a given extraction request. These parameters influence when one access method will be used over another (I.E.; when a **Read Logical** will be generated over a **Find**, or a **Read Physical** should be generated over anything else).

With the above information captured, NatQuery is provided with “application intelligence”; an application intelligence that can be compared to a skilled Natural programmer who understands the nuances of a given application file structure. This captured intelligence can then be utilized on demand to enable the End-User to generate optimized data extract programs upon request.

This approach shields the end-user from the complexities of how source data is physically stored, how that data is physically retrieved and extracted, the manner in which data gets moved back to a workstation, or the integration method used to move desired data into a user designated target.

Creating a NatQuery Configuration

The process of building a NatQuery Configuration from "scratch" involves 6 steps, with each step having its own sub-steps. The general steps required would be:

1. **General NatQuery Information**
   General NatQuery Configuration provides basic information to a NatQuery installation. This process will identify the initial user of NatQuery, and will establish (among other things) the "**Environment Path**" that will become the directory into which a full Environment Configuration will be built by subsequent steps.

   Please refer to the section entitled: Step 1 – General NatQuery Configuration.

2. **Server Connection Information**
   Server Connection Information captures specific information on how Natural on the server platform is configured, so it captures the method that NatQuery will optionally use to connect to the Natural server platform, in addition to further information as to how the connection to the Natural server platform will be configured.

   Please refer to the section entitled: Step 2 – Server Connection Information.

3. **User Information**
   User Information defines the users of NatQuery, and establishes the attributes of these users.

   Please refer to the section entitled: Step 3 – User Information.

4. **Initial JCL / Script Templates**
   JCL / Script templates define how NatQuery generated processes will be physically executed within the Batch environment on the Natural Server.

   Please refer to the section entitled: Step 4 – Create Initial JCL / Script Templates.

5. **Server Platform Initialization**
   To facilitate NatQuery's interaction with the Natural Server Platform, certain installation-provided Natural programs need to be introduced into the Natural environment.

   Please refer to the section entitled: Step 5 – Server Platform Initialization.

6. **Environment Configuration**
   With the above information provided, NatQuery should be able to interact with the remote Natural Server environment so that knowledge about the ADABAS environment can be supplied. This primarily revolves around capturing DDMs for the files that will be extracted from, as well as capturing information about these DDMs (such as Occurrence Information, Descriptor Statistic Information and File Relationship Information).

Please refer to the section entitled: Step 6 – Environment Configuration.

Performing the instructions relating to each of these 6 steps will provide NatQuery with the information required to allow NatQuery to intelligently generate extracts against a specific ADABAS source environment.

The process of creating an **Environment Configuration** begins with handling General NatQuery Configuration as described in the following section.

Step 1 - General NatQuery Configuration

To access the **NatQuery Configuration** function, start NatQuery (if not already started) and on an empty NatQuery desktop click the **Administer** drop-down menu, then click on **NatQuery Configuration**. These actions will present a window named **NatQuery Configuration**.

The **NatQuery Configuration** window is comprised of multiple "tabs" and by default the **User Identification** tab will be selected.

1. **User-Identification tab**
   The **User Identification** tab presents a single field, **Server User ID** along with a large button labeled as "**Check Server for Support of Entered Server User ID**".

   The **Server User ID** field identifies the primary user of a given NatQuery installation, and this value will be specific to the installation machine unless subsequently changed. The **User ID** value entered should be the User ID of the person performing the install, and it should otherwise also be a User ID that will allow the user to access the server platform.

   For example, if NatQuery will be using **FTP** to connect to the Natural server, then the **User ID** entered will need to be the **User ID** value that allows the user to perform an FTP operation against the Natural server. If NatQuery will use **PC Network** operations to connect to the Natural server, then the **User ID** entered will be a **User ID** value that has access permissions against the Natural server.

   Enter the value of your **User ID** that will support a connection to the Natural server platform (either a defined FTP User if FTP integration is intended, or a defined network user if PC Network integration is intended).

   Now click on the **Environment Paths** tab, and proceed to the next step.

   NOTE: On the bottom of the **User Identification** tab there is a button labeled as "**Check Server for Support of Entered Server User ID**". Once NatQuery basic NatQuery configuration has been completed, then this button can test a user's connectivity to the remote server however at this point in the configuration process - using this button will produce an error (integration information has not yet been configured).

2. **Environment Paths tab**
   The **Environment Paths** tab captures three pieces of information; the **Output Path**, the **Query Path** and the **Environment Configuration Path**.

   Of these three paths, the most critical path is the **Environment Configuration Path**.

   2.1. **Output Path**
   The **Output Path** is the directory where NatQuery will place data and objects retrieved from the server. In some cases, this will also be the directory which NatQuery may generate objects into.

By default, the **Output Path** will default to the **Output** sub-directory of the NatQuery directory created under the user's Profile Directory (the full path to this directory will show at the bottom of the Environment Paths tab and will be labeled as "Working Path".

It is recommended to accept the provided default; I.E. the **Output Path** should be set so that it points at the **Output** sub-directory of the NatQuery install directory. This will usually be:

C:\Users\user-id\NatQuery\Output\

...where "***user-id***" is the value of the User ID supplied in Step 1 above.

Changing the **Output Path**’s text value can most easily be accomplished by using the **Browse** button immediately to the right of the **Output Path** text field. Clicking this button will invoke the **Select Output Path** window that will allow the user a graphical selection to their choice of directories.

2.2. **Query Path**
The **Query Path** is the directory where NatQuery will save query specifications, as well as the directory that previously saved queries may be opened from.

By default, the **Query Path** will default to the Demo sub-directory of the NatQuery installation directory.

It is recommended that the **Query Path** should be changed to the **Queries** sub-directory of the NatQuery install directory; this will usually be:

C:\Users\user-id\NatQuery\Queries\

...where "***user-id***" is the value of the User ID supplied in Step 1 above.

Changing the **Query Path**’s text value can most easily be accomplished by using the **Browse** button immediately to the right of the **Query Path** text field. Clicking this button will invoke the **Select Query Path** window that will allow the user a graphical selection to their choice of directories.

By setting the **Query Path**, queries that are built subsequent to the completion of the configuration process will be separated from the sample DEMO queries provided with an install.

2.3. **Environment Configuration Path**
This path is the most critical of the three paths displayed, as it is the **Environment Configuration Path** that NatQuery uses to store and retrieve information that pertains to a specific Natural / ADABAS source environment.

By default, an installation of NatQuery will set the **Environment Configuration Path** so that it points to the **Demo** sub-directory of the NatQuery installation directory.

For the initial installation, the **Environment Configuration Path** should be changed to point at the **Files** sub-directory of the NatQuery installation directory; this will usually be:

C:\Users\user-id\NatQuery\Files\

...where "***user-id***" is the value of the User ID supplied in Step 1 above.

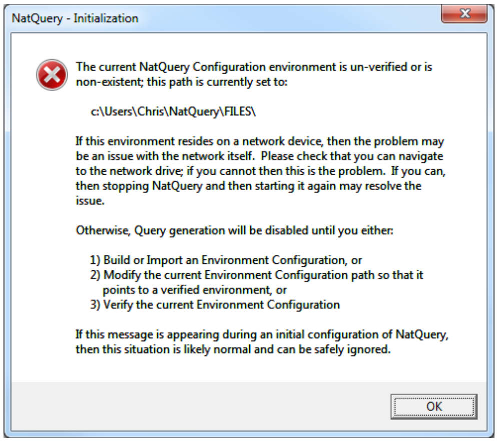
Changing the **Environment Configuration Path** can most easily be accomplished by using the **Browse** button immediately to the right of the **Environment Configuration Path.** Clicking this button will invoke the **Select Environment Path** window that will allow for graphical selection of the **Files** sub-directory.

The remaining tabs of the **NatQuery Configuration** function may be explored and their respective contents modified at this time, however with the exception of the **License Key** tab, the remaining tabs are for general default values. While this information is important to the operation of NatQuery, it is not critical to the building of a **NatQuery Environment Configuration**.

To speed the initial configuration task, *it is recommended* that the information handled through the other tabs be ignored at this time.

To continue with configuration, the **NatQuery Configuration** window can be closed by clicking the **OK** button located at the bottom of the window.

When the NatQuery Configuration window closes, and assuming that the above instructions have been followed, NatQuery will now produce an error message box similar to Figure 2 following:



The current NatQuery Configuration environment is un-verified or is non-existent; this path is currently set to:

c:\Users\Chris\NatQuery\FILES\

If this environment resides on a network device, then the problem may be an issue with the network itself. Please check that you can navigate to the network drive; if you cannot then this is the problem. If you can, then stopping NatQuery and then starting it again may resolve the issue.

Otherwise, Query generation will be disabled until you either:

1) Build or Import an Environment Configuration, or
2) Modify the current Environment Configuration path so that it points to a verified environment, or
3) Verify the current Environment Configuration

If this message is appearing during an initial configuration of NatQuery, then this situation is likely normal and can be safely ignored.

Figure 2 – Un-Verified Environment

This error message is produced as a direct result of changing the **Environment Configuration Path** to a path (the **Files** sub-directory) that does not contain the information that is expected to be present in a fully usable Environment Configuration.

This situation is normal, as the necessary information will be created / entered in subsequent steps; therefore, the user should click **OK** to this message.

Immediately after clicking the **OK** button on the previous message, NatQuery will respond with a second error message similar to that seen in Figure 1 - User File Not Found.

The User File Not Found message is also produced as a direct result of changing the **Environment Configuration Path**, and indicates that NatQuery cannot find the internal file that contains information relating to the users of NatQuery that should normally exist in a fully defined Environment Configuration. So, this message is normal at this time and the user may click the **OK** button to continue.

These actions should return the user to the NatQuery desktop; note that there will be no toolbar icons active.

The user may now proceed to the next section entitled Step 2 – Server Connection Information.

Step 2 - Server Connection Information

The **Administer Server Connection Information** function allows information to be captured that relates to how connections will be made to the remote server platform.

To access the **Administer Server Connection Information** function, the user should click the **Administer** drop down menu, click **Environment Configuration**, click **Server Connection Information**, and then click on **Server Information**. These actions will invoke the **Administer Server Connection Configuration** window.

The **Administer Server Connection Configuration** window display nine (9) tabs; some of these tabs will become disabled due to not being relevant depending on how the configuration proceeds and various configuration options are selected.

When initially presented, the **Administer Server Connection Configuration** window will display the **General Connection Info** tab by default.

1. **General Connection Info tab**
   The **General Connection Info** tab captures information concerning the platform that Natural resides on, which will be the platform that NatQuery connects to.

   1.1. **Server Communication Mode**
   The **Server Communication Mode** controls how / if **Automated Movement** will occur between NatQuery and the Natural server platform. The options available are **none**, **FTP** and **PC Network**.

   If **FTP / FTPS** will be used to move objects between NatQuery and the Natural Server platform, then the user should select **FTP**.

   If automated File Copy operations will occur across a **PC Network**, then the user should select **PC Network**.

   If no integration with the remote Natural platform is desired then the **None** option can be selected, however if this is the case then **Note 3** below should be reviewed.

   Note 1:
   At the current time, **FTP / FTPS** integration is supported against all targets, and in most cases users will select **FTP** as the **Server Communication Mode**.

   Note 2:
   In specific situations, for example when the remote NATURAL / ADABAS server resides on Windows or on UNIX / Linux, and there is shared disk between the UNIX environment and the NatQuery installation machine, then **PC Network** may be selected. If **PC Network** is selected, then objects will be moved between the NatQuery platform and the Natural server platform with automated network file copy operations.

Note 3:
If a situation exists where neither **FTP** nor **PC Network** is believed to be appropriate, it is still generally advisable to set the **Server Communication Mode** to **FTP**. Setting this option will then fully support the generation of required JCL or Script. Prior to initiating an FTP operation (that may not be physically possible); NatQuery will request permission from the user to perform the **FTP** operation, which can then be cancelled, leaving all generated objects available for manual handling.

1.2. **Server Type**
The **Server Type** combo box allows the user to select the type of platform that Natural resides on. Under current versions of NatQuery, the available choice of Server Type will be **MVS**, **VSE**, **UNIX** or **WINDOWS**.

Each of these **Server Types** are "general" categories, and while the precise **Server Type** may not be listed, a selection should be made of the **Server Type** that is closest to the type of the Natural server platform. For example, the **MVS** type covers all MVS systems, the **UNIX** category would cover all flavors of UNIX including Linux, and **WINDOWS** covers all Windows-based systems.

After selecting the appropriate **Server Type**, the user will note that one or more tabs of the **Administer Server Connection Information** tab may immediately become disabled. This will be because the selected **Server Type**, in conjunction with the previously selected **Server Communication Mode**, will make some of the tabs irrelevant to some configuration options.

1.3. **Server Name**
The **Server Name** identifies the machine that Natural server resides on.

When **FTP** is the selected **Server Communication Mode**, then the value entered here should be the IP address of the Natural server machine, or the Universal Resource Locator (URL) for this machine. For the FTP operations initiated by NatQuery to work successfully, this is a critical piece of information.

When **PC Network** is the selected **Server Communication Mode**, then the value entered here should be set to the machine name of the Natural server machine. While a value is required for this field, the value entered here is non-critical since copy operations will occur using a mapped drive or through a Universal Naming Convention (UNC), with this information being provided on another tab.

When none is the selected Server Communication Mode, this field will be disabled.

1.4. **Extract Request Submission Options**
If the selected **Server Communication Mode** is **FTP** or **PC Network**, then an **Extract Request Submission Options** frame will be made visible which will contain the options available that will inform NatQuery how (or if) generated JCL / Scripts will be executed on the Natural server platform.

The text of the options displayed in the **Request Submission Options** frame will vary depending on the setting of **Server Communication Mode** (**FTP** or **PC Network**) and the setting of **Server Type** (**MVS**, **VSE**, **UNIX** or **NT**).

Based on the value of the previously selected **Server Communication Mode**, the Administrator will proceed to the following section entitled **Server Communication Mode is FTP**, or should alternatively proceed to the subsequent section entitled **Server Communication Mode is PC Network**.

1.4.1. Server Communication Mode is FTP

When **FTP** is the selected **Server Communication Mode**, then the text associated with the two displayed radio buttons in the **Request Submission Options** frame will begin with either "**Direct FTP …**" or "**Just FTP…**".

Based on the discussions that follow in the sections entitled **Direct FTP** and **Just FTP**, the Administrator will select the appropriate option in the **Request Submission Options** frame.

1.4.1.1. Direct FTP

If the **Server Type** is **MVS** or **VSE**, then **Direct FTP** will result in generated JCL / Scripts being FTPed directly into JES (for MVS) or POWER (for VSE) on the remote mainframe, thereby enabling automatic execution against these platforms.

If the **Server Type** is **UNIX** or **NT**, then **Direct FTP** will result in generated Scripts and related objects being placed into files on the server via automated FTP. NatQuery will then attempt to invoke a Remote Execution utility (defined later) that will perform the remote execution of the generated Scripts on the Natural server platform. Examples of Remote Execution utilities that might be used would be RSH or REXEC, or third-party remote execution software solutions such as PLINK (www.putty.org) or PSEXEC (www.sysinternals.com).

1.4.1.2. Just FTP

How NatQuery will react to a setting of **Just FTP** is consistent across all target Natural server platforms.

When set to **Just FTP**, NatQuery will move generated JCL / Scripts (and any related objects) to the Natural server platform with automated FTP, but once these objects are placed there NatQuery will make no attempt to execute this JCL or script.

Options for creating **Automated Execution** of NatQuery generated requests placed onto the server by **Just FTP** integration are discussed in a subsequent section entitled **NatQuery Server Configuration Steps**.

1.4.2. Server Communication Mode is PC Network

When **PC Network** is the selected **Server Communication Mode**, then the radio buttons in the **Request Submission Options** frame will have associated text that contains "**Copy Scripts…and Execute"** (referred to as "**Direct Copy**"), or "**Just Copy…"**.

Based on the discussions that follow in the sections entitled **Direct Copy** and **Just Copy**, the Administrator will select the appropriate option in the **Request Submission Options** frame.

Note:
PC Network as a **Server Communication Mode** is only supported against a **Windows** or **UNIX** server platforms where shared disk is available.

1.4.2.1. Direct Copy

When set to **Direct Copy**, NatQuery will automatically move generated Scripts and related objects onto the server via automated network file copy operations. NatQuery will then attempt to invoke a Remote Execution utility that will perform the remote execution of the generated Script. Examples of Remote Execution utilities that might be used would be SSH, RSH or REXEC, or third-party remote execution software solutions such as PLINK (www.putty.org) or PSEXEC (www.sysinternals.com).

1.4.2.2. Just Copy

With **Just Copy** NatQuery will automatically move a generated script and related objects to the Natural Server platform with an automated network Copy operation; but it will not attempt to then execute that script.

This approach will therefore support manual execution on the Natural server unless "poll and submit" software such as OPALIS (www.opalis.com) or similar software that monitors a specified directory and will handle / execute scripts at intervals or as they arrive.

Options for creating automated execution of NatQuery generated requests placed onto the server by **Just Copy** integration are discussed in a subsequent section entitled **NatQuery Server Configuration Steps**.

2. FTP Configuration tab

If the user has specified "FTP" as the **Server Communication Mode**, then the **FTP Configuration Tab** will be enabled and this tab should now be selected by clicking on this tab.

2.1. FTP Information Frame

The FTP Information frame contains configuration information that is specific to FTP operations.

2.1.1. **Encryption**

The value of the **Encryption** selection box should be set to the type of FTP connection that is desired. Options are “**None** (Normal FTP)”, “**Implicit** FTPS” and “**Explicit** FTPS”; select the type of FTP communication that matches the setting of the FTP Server on the remote NATURAL / ADABAS platform.

If basic FTP operations are desired, then “**None** (Normal FTP)” should be selected.

If the use of FTPS (FTP over SSL) is required, then the user will select either **Implicit** or **Explicit** FTPS.

The **Explicit** method is a legacy compatible implementation where FTPS-aware clients can invoke security with an FTPS-aware server without breaking overall FTP functionality with non-FTPS aware clients.

The **Implicit** method requires that all clients of the FTPS server be aware that SSL is to be used on the session, and thus is incompatible with non-FTPS aware clients.

If FTPS is used, then the Client Machine will need to have Secure FTP Certificate(s) provided to complete the Secure FTP configuration against the target platform; the instructions for handling this Certificate can be found in the section of this manual entitled **Handling Secure FTP Certificates**.

2.1.2. Port

This should be set to the value of the **Port** on the remote FTP Server that will handle the FTP Connection. For normal FTP, this is usually “**21**”, and in most cases will be “**990**” when FTPS Implicit or Explicit connections are used.

Accept or change the **Port** value as needed.

2.1.3. Passive FTP

If checked, this checkbox will enable **Passive** (PASV) FTP communication. If left unchecked **Passive** FTP communication will be disabled.

The choice of using **Passive** versus **Active** FTP communication will be typically dictated by the server's FTP configuration, with **Active** FTP being the older of the two protocols. With **Active** FTP, the client machine initially specifies which client-side port it has opened for the data channel, and the server then initiates that connection. This contrasts to **Passive** FTP, where the server machine initially specifies which server-side port the client should connect to and the client then initiates the connection.

Usually, **Passive** FTP should be checked.

2.1.4. Create FTP Logfile

This checkbox controls whether or not a Log File of FTP Operations is created when a user performs a FTP operation. This Log File is a simple text file which can be useful when debugging FTP connection issues, but should be disabled (unchecked) when FTP operations are working properly. This is because a user's FTP password **IS** recorded in the Log File, and thereby represents a security risk if left enabled.

When checked, a log file is created with the name of:

userid_FTP_Trace.Log

This file is created in the path specified by the **NatQuery Environment Path**, where "***userid***" is replaced with the User ID of the NatQuery user.

2.1.5. FTP Establishes Remote Working Directory

When an FTP connection is made to a server, the FTP server configuration will determine the directory which FTP will initially connect to for each user. If the directory that FTP connects to is not where NatQuery should place objects and execute objects from, then once the FTP connection is established, FTP will have to issue a Change Directory (CD or CWD) command to the correct directory.

If this checkbox is checked, then NatQuery will place and execute objects from the directory that FTP connects to by default.

If this checkbox is not checked, then once NatQuery initiates an FTP connection, NatQuery will attempt to perform a Change Directory command to the appropriate directory through FTP once the connection is established, with the value of the directory to change to being set on the **Directory References** tab using the User File Directories On Server text fields.

2.1.6. Delete Extract File from Server after Download

If this checkbox is checked, then after NatQuery has downloaded an extract file from the FTP server to the FTP Client workstation, NatQuery will issue a DELETE against the remote file using FTP – thereby freeing up disk space on the server.

If this checkbox is not checked, then the source file will remain on the server subsequent to the extract file being downloaded by FTP.

It is suggested that this option be checked so that disk space may be saved on the server platform.

2.1.7. Ignored SSL Certificate Errors Frame

When using Implicit or Explicit FTP and a Secured FTP connection is established, handshaking occurs that verifies the credentials of both the Client and the Server such that things as the Date of the Certificate, the Host Name in the Certificate and

"Root" path are checked to see if they are trusted.

Under normal circumstances Certificate Errors should NOT be ignored, such that a FTP should not be allowed if any such error occurs. To allow for continued operation even in the face of such errors however, selecting the appropriate checkbox or checkboxes will cause NatQuery to ignore such errors and continue processing the FTP requests.

Usually, these checkboxes should all be left unchecked, with these checkboxes only being enabled in Explicit or Implicit FTP is used (they will be disabled if no encryption is used).

2.1.8. FTP Command Strings Frame

The FTP Command Strings Frame allows for FTP Command strings to be inserted into FTP communications as General Commands (which will always be inserted) or PUT Command (which will be inserted as an FTP Command just before a PUT Operation is attempted).

Usually all FTP Command strings can be left empty.

3. PC Network / UNIX tab

If the Server Type is **MVS** or **VSE**, then the **PC Network / UNIX** tab will be disabled, and the user may proceed to the step 4.

The **PC Network / UNIX** tab controls various aspects of how NatQuery objects will interact with the remote Natural server, and captures information that are applicable when **Server Communication Mode** is either **FTP** or **PC Network**, and the **Server Type** is not **MVS** or **VSE**.

When handling the construction of the parameters on the **PC Network / UNIX** tab, the user should understand that many of these parameters are typically built using references to Dynamic Substitution Variables that allow for various values to be substituted by NatQuery into these parameters.

The available dynamic substitution variables typically used within the fields on this window are referenced in the drop down selection box named **Available Dynamic Substitution Variables** located at the bottom of the current tab. When a substitution variable is needed as seen in this drop-down list, the user may select the appropriate variable by clicking on it in this list to highlight it, then click Ctrl-C (hold down the "Ctrl" key, then press the "C" key and then release both keys). This action will copy the highlighted text string onto the internal Windows clipboard. This copied value can then be pasted into the appropriate spot by placing the cursor in the desired location and then using Ctrl-V (hold down the "Ctrl" key, then press the "V" key and then release both keys). This action will paste the just copied value into the text string at the cursor location. Alternatively, a substitution variable displayed in the **Available Dynamic Substitution Variables** drop-down list box may be manually entered (note use of all upper case).

Note:
The Dynamic Substitution variables shown in the **Available Dynamic Substitution Variables** drop-down list box are a subset of the available dynamic substitution variables that NatQuery can handle. For a complete list of all Dynamic Substitution Variables as well as definitions for the subset of Dynamic Substitution variables made available in the **Available Dynamic Substitution Variables** drop-down list, please refer to the section entitled NatQuery Dynamic Substitution Variable Reference Table.

3.1. Method of Execution frame

The **Method of Execution** frame determines how any execution on the Natural Server platform will be handled.

When the **PC Network / UNIX** tab is first presented to a user, NatQuery will default the radio buttons pertaining to **Remote Execution** or **Execution by Server Process** to be set based on the value of **Request Submission Options** as set in 1.4. If the **Request Submission Option** was set to **Direct FTP** or **Direct Copy**, then **Remote Execution** will be selected; if the **Request Submission Option** is set to **Just FTP** or **Just Copy**, then **Execution by Server Process** will be selected**.**

If the intention is to configure NatQuery so that it will move generated objects to the Natural Server and then use a remote execution utility to remotely execute these objects, then **Remote Execution** should be selected.

If the intention is to configure NatQuery so that NatQuery will move objects to the server where these processes will then be executed manually, or some other additional software will then execute these processes, then select **Execution by Server Process**.

It should be noted that the setting of the **Request Submission Option** (**Connection Info** tab) in 1.4 above directly effects the setting of the **Remote Execution** or **Execution by Server Process**. If the **Request Submission Option** was set to **Direct FTP** or **Direct Copy**, then NatQuery will automatically default to **Remote Execution**. Likewise, if the **Request Submission Option** was set to **Just FTP** or **Just Copy**, then NatQuery will automatically default to **Execution by Server Process**. If the user switches between **Remote Execution** and **Execution by Server Process**, this switch will cause NatQuery to automatically change the setting of **Request Submission Option** on the **Connection Info** tab to stay in sync with the new choice.

3.1.1. Remote Execution

If **Remote Execution** is selected, then the text box labeled **Command String for Remote Execution** will become enabled. This textbox is then used to capture the command string that will invoke the appropriate remote execution utility, as if this utility were getting executed from a Windows Command Prompt. What is entered here is just the portion of the command that invokes the remote execution utility, leaving off that portion of the final command that specifically identifies the file that will be executed.

To explain this differently, NatQuery will do the following when instructed to perform a remote execution: He will build the appropriate script and move this script onto the server using an FTP or a Copy operation. He will then take any string entered into the **Command String for Remote Execution** text box, and append onto the end of this string the name of the Natural batch script file that was just placed onto the server, thus creating a complete command string. NatQuery will then "shell" this command string to the Windows operating system, where it will be executed.

To assist in a understanding how to construct the **Command String for Remote Execution**, the following examples may prove helpful:

Example #1 – UNIX Server and RSH / REXEC Utility
If a UNIX server is the target and the RSH (Remote Shell) or REXEC (Remote Execution) utility will be used by NatQuery to remotely execute a NatQuery-generated process on the server (with this process having been placed onto the server using an automated FTP or Copy operation). The full command line string to accomplish this would generally be:

RSH ***hostname*** -l ***username*** -n ***command***
or
REXEC ***hostname*** -l ***username*** -n ***command***

In this case, the string entered into **Command String for Remote Execution** would be:

RSH ***hostname*** -l &&USER-ID –n
or
REXEC ***hostname*** -l &&USER-ID –n

In the above example, the value of ***hostname*** would be the actual name of the UNIX machine that hosts Natural and the RSH or REXEC Service. The value of ***username*** is changed to &&USER-ID, which is a dynamic substitution value that NatQuery will replace with the user ID of the user who invokes a remote execution. The ***command*** portion is completely removed, because NatQuery will append the appropriate command (which is the execution of Natural using a generated script) to the text string supplied when the remote execution physically occurs.

Note:
If the server authenticates the user to use RSH or REXEC, then the above examples would change to accomplish the passing of a network User-ID and related password. In this case, references to the dynamic variables &&NETWORK-USER and &&NETWORK-PASSWORD might be used, as these would allow NatQuery to prompt the user for these values and then substitute these values into the full command string prior to physically executing the full command string.

Example #2 - UNIX Server and the PLINK Utility
If a UNIX server is the target and the PuTTY utility is available, then the PLINK command of the PuTTY install canbe used by NatQuery to remotely execute a NatQuery-generated process on the server (with this process having been placed onto the server using an automated FTP or Copy operation). The full command line string to accomplish this would generally be:

PLINK -batch -l *username* -pw *userpassword hostname* sh

In this case, the string entered into **Command String for Remote Execution** would be:

PLINK -batch -l &&NETWORK-USER -pw &&NETWORK-PASSWORD ***hostname*** sh

In the above example, the value of ***hostname*** would be the actual name of the UNIX machine that is the server. The value of ***username*** is changed to &&NETWORK-USER and the value of ***userpassword*** is changed to &&NETWORK-PASSWORD, both of which NatQuery will prompt the user for and then substitute. The "sh" command is to shell the script name for execution on the remote server, with NatQuery supplying the name of that script at submission time.

Example #3 – Windows Server and PSEXEC Utility
If a Windows server is the target and the PSEXEC utility (www.sysinternals.com) is used to achieve remote execution, the command to accomplish this would be:

psexec ***hostname*** -u ***username*** -p ***userpassword*** -i ***command***

In this case, the string entered into **Command String for Remote Execution** would be:

psexec ***hostname*** -u &&NETWORK-USER -p &&NETWORK-PASSWORD -i

In the above example, the value of hostname would be the actual name of the Windows machine that hosts Natural, entered as a UNC name (example: \\strider). The value of ***username*** is changed to &&NETWORK-USER and the value of ***userpassword*** is changed to &&NETWORK-PASSWORD, which are dynamic substitution values that NatQuery will replace with values that will be prompted from the actual user. The ***command*** portion is completely removed, because NatQuery will append the appropriate command (which will be the name of a NatQuery-generated script file) to the text string supplied when the remote execution occurs.

Example #4 – Windows Server on Same Machine
If a Windows server is the target and this happens to be the same machine that

NatQuery is installed on, then the **Command String for Remote Execution** would be left empty. This would mean that the "remote" execution command being shelled to Windows would actually just be the name of a NatQuery-generated batch Natural script, which would then be executed locally.

3.1.2. Execution by Server Process

If **Execution by Server Process** is selected, then the text box labeled **Client Path to Server Pickup Directory** will become enabled. This text box is then used to capture the path from the NatQuery machine to the directory on the Natural Server machine into which NatQuery generated objects will be copied. This may be a "mapped" drive from the client machine to the server machine, but can also be a UNC reference.

To assist in understanding how to construct the **Client Path to Server Pickup Directory**, the following examples may prove helpful:

Example #1 – Windows Server – UNC
Assuming that Natural is installed onto a machine named STRIDER, and it has been decided that the directory where NatQuery should place generated objects on STRIDER is a directory named NATQUERY, then the appropriate setting for **Client Path to Server Pickup Directory** would be:

\\STRIDER\NATQUERY\

Example #2 – Windows / UNIX Server – Mapped Drive
In this case, the user will create a Mapped Drive using the Tools drop-down menu of Window Explorer (or alternative method), and then reference that mapped drive in the **Client Path to Server Pickup Directory**. Therefore, if a Mapped Drive named "Z:" is created to point at the directory on the server machine where it is intended that NatQuery place generated objects, then **Client Path to Server Pickup Directory** would be set to:

Z:\

Example #3 – Windows Server on Same Machine
In this case, the value of **Client Path to Server Pickup Directory** would be set to the directory on the local machine where the NatQuery generated objects should be placed.

3.2. Path Definitions frame

The Path Definitions Frame captures path information as to how NatQuery or NatQuery generated processes will reference directories on the Natural Server. There are two such references: How the NatQuery client machine will refer to the appropriate directory on the server machine (I.E., the **Client's Path to Server Work Directory**), and how a process being executed on the server machine will refer to the execution directory (I.E., the **Server's Path to Server Work Directory**).

Depending on the selected **Server Communication Mode** (**FTP** or **PC Network**) and the selected **Method of Execution** (**Remote Execution** or **Execution by Server Process**), either one or both text boxes may be disabled. Enter the values into the available text boxes as follows.

3.2.1. Client's Path to Server Work Directory

The **Client's Path to Server Work Directory** will be available in the situation where **Server Communication Mode** is set to **PC Network** and **Method of Execution** is set to **Remote Execution**; in all other situations, this text box should be disabled.

The value entered into this field will be the UNC name or mapped drive from the NatQuery installation machine to the directory on the Natural server machine where NatQuery will place generated objects.

3.2.2. Server's Path to Server Work Directory

The **Server's Path to Server Work Directory** will be available in a number of situations.

The value entered into this field will be the Server's path to the directory on the Server into which NatQuery will be placing generated objects.

When **Remote Execution** is being used, then the command being issued to perform the remote execution needs to be given the path to the script file to execute, in addition to the name of the script itself.

For example, it may be noted that the **Remote Execution** command referenced above does not specify the name of the script file to be executed - the name of the script file is generated by NatQuery. Likewise, the relative path to that script file is also not part of the Remote Execution command. The value entered into **Server's Path to Server Work Directory** supplies this path value, such that prior to shelling the **Remote Execution** command, the value entered into **Server's Path to Server Work Directory** is concatenated to the value of the script file name, with the resulting concatenated string then being appended by NatQuery to the end of the **Remote Execution** command to form a complete **Remote Execution** command string which can then be shelled.

3.2.3. Server Log Copy Command

The **Server Log Copy Command** is used by a generated NatQuery extract program to handle the proper updating of a user's Log File within an Open System environment.

The necessity of the **Server Log Copy Command** is due to how Natural handles Work Files in UNIX and Windows platforms, and the need to have the current contents of a User's Log File retained when the user's Log File is updated to reflect the completion of new a request.

As of this writing, if Natural on Open System is pointed at an existing file that currently contains data as a Work File, and then Natural writes to this Work File, the existing contents will be overlaid.

To properly retain the existing contents of a User's Log File, and then append onto this file the execution status of a newly submitted request, the script that executes the user's request will first write the execution status of the just-executed request to a temporary Log File. After writing the temporary Log File, the NatQuery-generated program will then Close that Work File, and the program will then shell the **Server Log Copy Command** to the server's operating system.

Through the use of the **Server Log Copy Command** then, any existing contents of a User's server-base "Remote" Log File can be retained while appending into this file the execution status of a just-executed request.

Upon initial configuration, NatQuery will provide a default value for **Server Log Copy Command**, and this will be done based on the initial selection of **Server Type** (set on the **Connection Info** tab in step 1.1).

Server Log Copy Command Against UNIX Systems - Discussion
For **UNIX** systems, the **Server Log Copy Command** needed to append the contents of one file (in our case, the ***temporary_log***) to the end of another (in our case, the ***permanent_log***) would be accomplished by a command similar to the following:

cat ***temporary_log*** >> ***permanent_log***

To allow for this command to be dynamically handled, the Natural programs generated by NatQuery will shell, through a User Exit, the command necessary to perform the cat operation, and it will utilize the Environment Variables of $CMWKF20 and $CMWKF21 (which will be pointed at the appropriate temporary and permanent User Log files by the script); this then allows the command needed to perform the cat. Therefore, on UNIX systems *it is recommended* that the value of **Server Log Copy Command** should be set to:

```
cat $CMWKF20 >> $CMWKF21
```

The above example **Server Log Copy Command** is set by default on the initial installation of NatQuery, and for most UNIX / Linux systems this command should require no changes.

Server Log Copy Command Against Windows Systems - Discussion
For **Windows** systems, the **Server Log Copy Command** needed to append the contents of one file (in this case, the ***temporary_log***) to the end of another file (in

this case, the ***permanent_log***) would be accomplished by a command similar to the following:

```
cmd.exe /c copy /b permanent_log + /b temporary_log /b permanent_log /b
```

To allow for this command to be dynamically handled, the Scripts that NatQuery will utilize in conjunction with internally issuing this command will contain references to the symbolic variables of CMWKF20 and CMWKF21 (which will be pointed at the appropriate temporary and permanent User Log files via the generated script), that then allow the command to become generic. Therefore, on Windows systems *it is recommended* that the value of **Server Log Copy Command** should be set to:

```
cmd.exe /c copy /b %CMWKF21% + /b %CMWKF20% /b %CMWKF21% /b
```

The above example **Server Log Copy Command** is set by default on the initial installation of NatQuery, and for most Windows systems this command should require no changes.

4. Natural Configuration tab

The Natural Configuration tab controls various parameters that describe Natural on the Server.

4.1. Natural Server Library

The **Natural Server Library** represents the value of the Natural library which NatQuery will primarily interact with when it connects to Natural on the server platform. The **Natural Server Library** text box represents a value associated with a dynamic substitution variable named **&&NATURAL-USER-LIBRARY**, with this value being usable in a JCL / Script templates whenever there is a need to dynamically reference the target Natural library (I.E. for LOGON purposes).

Unless another Natural library will be used to contain NatQuery's programs, *it is suggested* the value **NATWORKS** should be entered into the **Natural Server Library** textbox; however any valid Natural Library name may be used, and this can either be a new Natural library or an existing Natural library.

Note:
If the Software AG product Natural Security is used on the Natural server platform, then the name of the **Natural Server Library** will need to be defined to Natural Security such that NatQuery users will be allowed to access this library. The user will be reminded in a subsequent step of this need, however if the library name will not be NATWORKS, then it may be a good idea to make a note of the name of the intended **Natural Server Library**.

4.2. Date Format

The **Date Format** represents how Date Format fields (I.E. DDM fields that have a format of “D”) will be output by NatQuery, as well as how NatQuery will prompt a user when inputting values used in conjunction with “D” format fields.

The Administrator will need to make a decision that is consistent for the site.

Note:
If there is any question as to the appropriate setting of the **Date Format** parameter, then this value should be set so that it corresponds to the Natural Parameter module (NATPARM) setting of the **DT** parameter.

4.3. Decimal Character

The **Decimal Character** is used to indicate the character that NatQuery should use when it needs to handle the output of a numeric field.

For countries where English is the predominant language, the typical setting for the **Decimal Character** would be the period character (“.”).

For most International settings, it is usual for the **Decimal Character** to be set to the comma character (“,”).

Note:
If there is any doubt as to the appropriate setting of the **Decimal Character**, then this value should be set so that it corresponds to the Natural Parameter module setting of the **DC** parameter.

4.4. Variable Prefix Character

The **Variable Prefix Character** is used by NatQuery as a prefix character on any variables that NatQuery may need to generate into a Natural program.

For countries where English is the predominant language, the typical setting for the Variable Prefix Character would be the number sign character (“**#**”).

For international settings, the user should enter the character that is generally used according to any local programming standards.

4.5. Command Delimiter Character

The **Command Delimiter Character** is used by NatQuery when it is necessary to handle multiple Natural commands and allow these commands to be logically separated / delimited.

The value of this parameter varies from site to site, but for countries where English is the predominant language, this value is typically the comma character (“**,**”).

Note:
If there is any doubt as to the appropriate setting of the **Natural Command Delimiter**, then this value should be set so that it corresponds to the Natural Parameter module

setting of the **ID** parameter.

4.6. Natural Security Installed

The **Natural Security Installed** checkbox is used to inform NatQuery that Natural Security logins must be properly handled when JCL or Scripts are generated that will run against the Natural server.

Essentially, if the **Natural Security Installed** checkbox is checked, then NatQuery will prompt the NatQuery User for the values of Natural Security User-ID and related Natural Security password, which will then become available for substitution into scripts at the appropriate points. This therefore allows scripts to be created that reference these substitution variables, with NatQuery then performing dynamic substitutions with the appropriate values when they are encountered.

If Natural Security is installed on the Natural Server platform, then it is suggested that the **Natural Security Installed** checkbox be selected.

If Natural Security is not installed on the Natural Server, or it is desired to have all NatQuery requests run under a single Natural Security User-ID, then it is advised to leave the **Natural Security Installed** checkbox un-checked.

NOTE:
When creating JCL / Script templates, values for Natural User ID and Password can be "hard-coded" into the templates however this is not usually recommended for security reasons.

4.7. ADABAS/NATURAL Multi-Fetch Support Frame

The **Multi-Fetch Support** frame controls information relating to the automatic use of Natural's Multi-Fetch / Pre-Fetch by NatQuery.

It is suggested that the Multi-Fetch option be left un-checked for the initial configuration, and therefore defer configuring Multi-Fetch to a later time after a working Environment Configuration has been established.

In order to "turn on" NatQuery's automatic handling of Multi-Fetch / Pre-Fetch, the user will first click the **Multi-Fetch Enabled** checkbox so that it is checked. Doing so will enable the **ADABAS Multi-Fetch** and **Natural Multi-Fetch** radio buttons.

The user will select the type of Multi-Fetch that is desired, and will then review the values associated with each.

Subsequently, when NatQuery has determined that, say, a Read Physical is the best choice for data access, the generated processing will automatically invoke the desired Multi-Fetch option.

5. **File References tab**
   The text fields captured on the **File References** tab pertain to a mechanism referred to as the "**NatQuery Server".** The **NatQuery Server** represents a method of remote execution against **MVS** or **VSE** platforms that is no longer actively supported by NatWorks, and it is available only for the purpose of backward compatibility.

   In all cases, a new installation of NatQuery should ignore the text fields on this tab; I.E. leave them all blank.

6. **Directory References tab**
   The **Directory References** tab will only be available if **FTP** is the selected **Server Communication Mode**.

   If **PC Network** is the selected **Server Communication Mode**, then the **Directory References** tab will be disabled, and the user can skip to step 9.

   For **FTP** users, the **Directory References** tab controls when an automated FTP Change Directory (**CD**) operation should be handled by NatQuery prior to handling the files that NatQuery may PUT on the Natural server platform or GET from the Natural server platform.

   The **Directory References** tab is divided into two frames, **User File Directories on Server** and **Server File Directories on Server**.

   6.1. **User File Directories on Server Frame**
   The text boxes in the **User File Directories on Server** frame allow for the optional capture of directory paths that specify where a given site may wish to have **User Request Files**, **User Output Files,** and **User Log Files** placed.

   The values that may or may not be required in the **Request Files Directory**, **Output Files Directory** and **Log Files Directory** text boxes are dependent upon whether or not FTP establishes a "working directory" in relation to where a site ultimately wants to place the files that NatQuery will interact with on the Natural server platform.

   With FTP, it is common for FTP to establish a "working directory" when an FTP connection is established against a remote server platform. In some cases this default working directory may be an acceptable location for where NatQuery should both **PUT** files into, as well as **GET** files from. In other situations however, the default working directory may not be satisfactory for resolving some or all of NatQuery's server-based files, thus necessitating the need for an FTP connection to perform a Change Directory (**CD**) operation from the default working directory established at connection time to a different directory prior to then performing a **PUT** or **GET** or other FTP operation against a specific type of object (**User Request File**, **User Output File** or **User Log File**).

   This is where the text boxes found in the **User File Directories on Server Frame** come into play. Any path value entered into these text fields will cause NatQuery to automatically issue a Change Directory (CD) command to the appropriate directory prior to handling a specified object type (a **User Request File**, a **User Output File,** or a **User's Log File**).

Based on the type of server platform that Natural is installed on, the Administrator should review the appropriate section that pertains to that server for further information on what the settings of the text boxes in **User File Directories on Server Frame** should be.

6.1.1. MVS Natural Server Platform

When NatQuery is used against an **MVS** Natural server platform, then in most cases the FTP server is configured such that a working directory is established by default, with the "working directory" equating to a high-level qualifier that is equal to the FTP user's User ID.

In situations where FTP does establish a working directory that equates to the User ID of the user initiating the FTP connection through NatQuery, then *it is suggested* that no values be entered into any of the **User File Directories on Server Frame.** By leaving all values blank, NatQuery will **PUT** into files that then inherit the default high-level qualifier, meaning that NatQuery files for a user will appear under each user's User ID as a high-level qualifier. Similarly, when NatQuery is instructed to **GET** a file from the server, the FTP operation will assume that the file upon which the **GET** will act will have a high-level qualifier of the initiating FTP user.

If a site does not want to have NatQuery-related files appear under individual User IDs and instead wishes to have these files placed under a common high-level qualifier(s), then the name of the desired high-level qualifier(s) should be entered into the corresponding path text boxes.

6.1.2. VSE Natural Server Platform

When NatQuery is used against a **VSE** Natural server platform, then the site will either be using a VSAM catalog to contain NatQuery files, a utility such as Dynam, or perhaps use straight sequential files.

If a **VSE** site will be using a VSAM catalog, then it is suggested that all files that NatQuery will interact with will be placed within the VSAM catalog. The FTP server on VSE will be configured to know of the existence of the VSAM catalog by having the name of the VSAM catalog defined in the FTP startup deck. In this case, the name of the VSAM catalog should be entered into each of the directory path fields under the **User File Directories on Server Frame**. If a VSAM catalog is to be used, then NatWorks suggests the name of "NATWORK", as subsequent steps will assume that this was the name used.

If a **VSE** site will be using a facility such as Dynam or straight sequential files to handle NatQuery's server-based files, then the Administrator must understand how an FTP connection does or does not establish a working directory in order to have NatQuery place and retrieve files correctly against the Natural server platform. If an FTP connection does establish a working directory (usually under a high-level qualifier that equates to the User's User ID), and it is acceptable to the Administrator that NatQuery-related server files will be stored under the respective User's User ID value, then no values need to be entered into the directory path fields in the **User File Directories on Server Frame**.

If a site does not want to have NatQuery-related files appear under individual User IDs and instead wishes to have these files placed under a common high-level qualifier(s), then the name of the desired -level qualifier(s) should be entered into the corresponding path text boxes.

6.1.3. UNIX, OpenVMS, or Windows Natural Server Platform

When used against a **UNIX, OpenVMS,** or **Windows** server platform, the directory values entered will equate to the directory where it is desired that NatQuery-created files should be placed.

To streamline configuration, *it is suggested* that NatQuery initially be configured so that all objects are resolved from the same directory on the Natural server platform. *It is therefore suggested* that a target directory named "NatQuery" be established for this purpose on the server platform. In this case, all three of the text boxes in the **User File Directories on Server** will be set to the same value, which will be the full relative path to the "NatQuery" directory.

If there is any question about what path value(s) should be placed into a directory path text box, it is encouraged that the Administrator manually establish an FTP session from a Windows Command Prompt against the remote server. By observing what is established as a default FTP directory once the Administrator Logs into FTP, the Administrator can then test the path value that may need to be entered into a directory path text box by using this path value in conjunction with the FTP command of "**CD"** (Change Directory).

6.2. Server File Directories on Server Frame

The text boxes in the **Server File Directories on Server** pertain to the use of the "NatQuery Server". The NatQuery Server is a method of remote execution against **MVS** or VSE systems that are no longer actively supported by NatWorks, with the text boxes in this frame being available for backward compatibility purposes only.

In all cases, a new installation of NatQuery should ignore the text fields in the **Server File Directories on Server** frame (I.E. leave them blank) and continue on to the **User File Directories on Server Frame** as described below unless otherwise instructed to use these fields by NatWorks.

7. Disk Size Constants tab

The **Disk Size Constants** tab will only be available if **FTP** is the selected **Server**

Communication Mode, and the selected **Server Type** is set to **MVS** or **VSE**.

If **FTP** is the selected **Server Communication Mode**, but the **Server Type** is **UNIX, OpenVMS,** or **Windows;** or if **PC Network** is the selected **Server Communication Mode**, then the user should skip to step 9.

For **MVS** or **VSE** servers, the **Disk Size Constants** tab contains two frames; one called **Physical Disk Attributes** and another called **Disk Allocation Calculation Factors**.

7.1. **Physical Disk Attributes Frame**

The **Physical Disk Attributes** frame informs NatQuery of the default settings for the disk drive(s) that will be used to hold extracted data. As a result of NatQuery generating an extract, NatQuery will calculate the Logical Record Length (LRECL) for the extract records, and NatQuery will then additionally calculate an Estimated Number of Records that are likely to be returned by the extract request.

Using the calculated LRECL and the Estimated Number of Records information, NatQuery can then calculate the type of Unit to request (Cylinders or Tracks), the number of Units required, as well as an optimal Block size by using the information provided in the **Physical Disk Attributes** frame. NatQuery will then dynamically substitute these values into JCL templates by having these templates reference the appropriate Dynamic Substitution Variable tags.

By default, NatQuery provides values for **Physical Disk Attributes** that describe an IBM 3390 disk drive. If the disk in use at a given site has different attributes, then the Administrator should enter the appropriate values for **Block Size**, **Blocks per Track**, **Tracks per Cylinder,** and **Cylinders per Volume** that accurately describe the disk that will be used.

When the settings for **Physical Disk Attributes** are correct, the Administrator can continue to the next step of this section - in most cases the default values supplied will work.

7.2. **Disk Allocation Calculation Factors Frame**

The **Disk Allocation Calculation Factors** frame controls how NatQuery should calculate the **Primary** and **Secondary** disk allocations that can be made by JCL in **MVS** or **VSE** environments.

The **Primary Disk Allocation Factor** is used by NatQuery to calculate the amount of disk units that should be requested as a **Primary Allocation** through JCL, and the **Secondary Disk Allocation Factor** is used to calculate the amount of disk units that should be requested as a **Secondary Allocation**.

For example; assume that NatQuery determines that a given request needs 10 Cylinders to hold the extracted data. By setting the **Primary Disk Allocation Factor** to “.5”, then NatQuery will calculate a **Primary Disk Allocation** of 5 Cylinders (.5 of the 10 total

Cylinders). By setting the **Secondary Disk Allocation Factor** to ".2", then NatQuery would calculate a **Secondary Disk Allocation** of 2 cylinders (with the Secondary Allocation occurring up to 15 times). Both of these calculated values can then be automatically inserted into JCL by NatQuery when it encounters the dynamic substitution variables that relate to these fields.

By using the **Disk Allocation Calculation Factors**, an Administrator can control how NatQuery allocates disk, and can provide "padding" for those situations where NatQuery may possibly under-estimate the total amount of disk required.

8. **Job Priority tab**
The **Job Priority** tab will only be available if **FTP** is the selected **Server Communication Mode** and the selected **Server Type** is set to **MVS** or **VSE**.

If **FTP** is the selected **Server Communication Mode** and the **Server Type** is UNIX, **OpenVMS**, **Windows,** or **PC Network** is the selected **Server Communication Mode**, then the user should skip to step 9.

When the server is **MVS** or **VSE**, the **Job Priority** tab allows the Administrator to specify the Job Class values that a user may submit NatQuery requests into. Job Classes can be specified as either a **High Priority** or **Low Priority** specification.

While Job Class values are required to be entered for both **High** and **Low Priority Execution**, these are optional in the sense that they may or may not be physically used (through dynamic references) in the JCL templates that will be configured later. The values entered may therefore be fictitious, and they may be set to the same or different values.

When a user indicates that they wish to submit a request for execution, they will be given the option of selecting **High Priority Execution** or **Low Priority Execution**. If a JCL template (built later) is constructed to utilize the appropriate Dynamic Substitution parameter that handles Job Class, and based on the high / low priority the user selects, the appropriate high / low job class will be substituted into the JCL stream that is then FTPed onto the server.

After reviewing and possibly modifying the elements on the **Job Priority** tab, click on the **Miscellaneous** tab.

9. **Miscellaneous tab**
The **Miscellaneous** tab controls ancillary aspects of **Server Connection Information**, and is only available when the **Server Communication Mode** is set to **FTP**.

The **Miscellaneous** tab contains a single checkbox: **Enable E-Mail Handling of Output Data.**

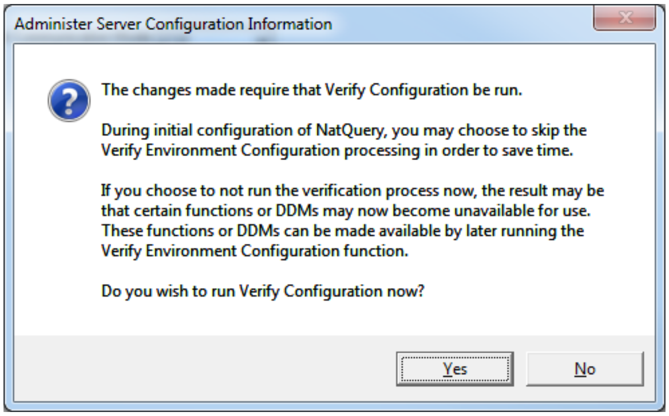
9.1. **Enable E-Mail Handling of Output Data**
The **Enable E-Mail Handling of Output Data** enables basic processing through which NatQuery can support the automatic e-mailing of data created via a NatQuery extract

request.

As this is an advanced feature, *it is suggested* that this checkbox be left unselected for initial configuration.

10. **Server Connection Information – Wrap Up**
At this point, the user should have completed all required information on the **Administer Server Connection Information** window, and this window can now be closed by clicking the **OK** button.

Immediately after clicking the **OK** button, the user will be presented with a message box that is similar to the following:



Administer Server Configuration Information

The changes made require that Verify Configuration be run.

During initial configuration of NatQuery, you may choose to skip the Verify Environment Configuration processing in order to save time.

If you choose to not run the verification process now, the result may be that certain functions or DDMs may now become unavailable for use. These functions or DDMs can be made available by later running the Verify Environment Configuration function.

Do you wish to run Verify Configuration now?

Figure 3 – Verification Question

The purpose of this message box is to inquire if the user would like to run a **Verify Configuration** process. The **Verify Configuration** process is a process that reviews all information that exists within the **Environment Configuration** for correctness and completeness. This process determines, among other things, what functions of NatQuery can be made available for use, and what DDMs are available for processing.

Since the **Environment Configuration** is in the early stages of being built, a verification of the current **Environment Configuration** at this point would result in numerous issues that will be addressed / resolved in subsequent configuration steps.

It is recommended that the user respond to the prompt shown in Figure 3 (**Verification Question**) by clicking the **No** button.

NatQuery will respond to this by issuing the error message as previously seen in Figure 2 (**Un-Verified Environment**), which can be ignored by clicking **OK**.

NatQuery will then issue an error message similar to what is seen in Figure 1 (**No Users Defined**), which can also be ignored by clicking **OK**.

As a point of information, the **Verify Configuration** process can be run at any time

NatQuery displays an empty desktop. This would be accomplished by clicking the **Administer** drop-down menu, then clicking **Environment Configuration,** and then clicking **Verify Environment Configuration**.

The user should now proceed to the next section entitled Step 3 - User Information.

Step 3 - User Information

The next step of the configuration process is to define the first user of NatQuery (who at this juncture is assumed to be an Administrator of NatQuery) as a NatQuery user.

Prior to creating the first user, *it is suggested* that a Default User Profile be created, with this Default Profile then being able to assist in the rapid setup of subsequent users. Creating a default profile will also assist in establishing consistent definitions across all users.

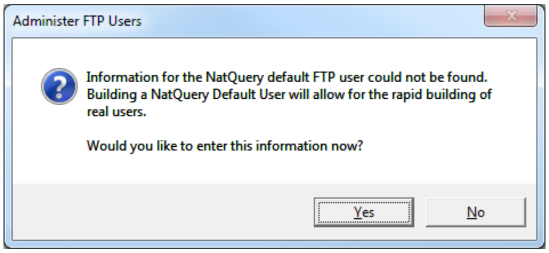
There are two steps to creating User Information:

- Step 3.1 – Create Default User Profile, and
- Step 3.2 – Create Initial User Profile.

Step 3.1 – Create Default User Profile

Both the Default Profile and User Profiles are handled using the **Administer User** function, which is invoked by clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration** > **Server Connection Configuration** > **User Information**.

When the **Administer Users** window is initially invoked, and as a result of there being no User Profiles currently existing, NatQuery will immediately present a message box similar to what is seen in Figure 4, which asks the user if they wish to create a Default User Profile.



Information for the NatQuery default FTP user could not be found. Building a NatQuery Default User will allow for the rapid building of real users.

Would you like to enter this information now?

Figure 4 - Default NatQuery User Prompt

To save time in the later definition of additional users, *it is suggested* that the Administrator should click **Yes** to this message box so as to create a Default User profile.

NatQuery will then present the **Administer User Information** window for the Default Profile with the following controls:

1. **Active Check Box**
   The **Active** check box indicates whether or not a new user should be defined as **Active** or **Inactive** to NatQuery. Only users flagged as **Active** will be able to submit extraction requests, so *it is suggested* that the **Active** check box be selected (which will be the default).

2. **Active Requests**
   The **Active Requests** text box indicates how many requests a user will be allowed to submit against the server at one time. The number shown here will correspond to "request slots", with any given "request slot" being re-usable and being able to contain a given specific extract request.

   By default, **Active Requests** will default to 3. Since the number of requests slots can be increased and decreased easily at a later time, *it is suggested* to keep the default value of 3.

   Note:
   The number entered for Active Requests will have a direct correlation to the number of files that are specified the in **File References** frame.

3. **File References Frame**
   The **File References** frame details the naming conventions that will be used for files used by NatQuery-generated processes on the Natural server platform. Specifically; the **File References** frame details how user **Request Files,** user **Output Files**, and each user's **Log**

File will be named.

For each type of file defined (**Request**, **Output**, or **Log**), NatQuery typically requires two references to the specific file. NatQuery requires how a given file will be referenced within generated JCL / Script (I.E. a **JCL / Script Reference**), and NatQuery requires a reference that will be used should the file need to be referenced for automatic movement between the NatQuery workstation and the Natural server platform (an **FTP Reference**).

While a given **JCL / Script Reference** and the corresponding **FTP Reference** will both point at the same physical file, the text value of these two references may be different.

When the Natural server platform is **MVS**, the **Server Communication Mode** is **FTP**, it has been indicated that **FTP Connection Establishes Working Directory**, and **Just FTP** has been specified; the **JCL / Script Reference** will usually include the User ID of the user submitting a request as a high-level qualifier, whereas the **FTP Reference** would not (the act of establishing an FTP connection would establish the high-level qualifier through the setting of **Working Directory**). In all other situations, it is typical that the **JCL / Script Reference** will be identical to the **FTP Reference**.

As will be seen, NatQuery will always provide defaults for required values, and these default values will be introduced based on the setting of **Server Type**. In most cases, the default values provided should be acceptable for initial configuration, and *it is suggested* that the default values be accepted as is.

In handling file names for the **Default User Profile,** the reader will note the use of several **Dynamic Substitution** variables, or text values that are preceded by two ampersands (“&&”). In providing default values for the Default User Profile, two Dynamic Substitution variables are referenced by default, **&&USER-ID** and **&&REQ-NUMBER**. The value of **&&USER-ID** will be automatically replaced by the ID of the user being created later (I.E. the **Default User Profile** will directly influence how a real user is defined). The **&&REQ-NUMBER** portion will be replaced by the number of the slot the request has been placed into. Non-highlighted text will be treated as text constants.

Note 1:
The **Default User Profile** will almost always reference dynamic substitution variables, as these variables will have suitable substitutions applied when a new user is subsequently added.

Note 2:
When the Natural server platform is **UNIX** or **Windows**, then the file references entered will only be the actual name of the file for both the **Output Files** and the **Log File**. The Request File reference will serve as a prefix value only. The names entered will not typically include any path information as path information will be embodied in Script templates.

Note 3:
Values provided into a Default User Profile provide the basis for how values will be set for a

new user being subsequently added. All default values can be overridden for each user, and can be changed at any time later.

3.1. Request File(s) tab

Request Files are the names of the files into which NatQuery-generated requests will be placed on the Natural server prior to execution. Each NatQuery user will have at least one **User Request File** specified, however the total number of User Request Files any given user will have is administratively determined (the default number will be three).

Note:
Request File information is required ***except*** in the case where **FTP** is the **Server Communication Mode** and **Direct FTP** is used against a mainframe Natural server. In these situations the **Request Files** tab will be disabled since JCL will not be placed in "files" per se, but directly placed into JES or POWER.

3.1.1. Mainframe Natural Server Platforms

For Mainframe Natural Servers, the following default names are suggested:

3.1.1.1. JCL / Script Reference

The default naming for the **Request File JCL / Script Reference** will be:

&&USER-ID.REQ**&&REQ-NUMBER**

3.1.1.2. FTP Reference

The default naming for the **Request File FTP Reference** will vary depending on the type of mainframe server.

For **MVS** Systems when the **FTP Session Establishes Working Directory** option is set, the default naming will be:

REQ**&&REQ-NUMBER**

In all other situations, the default naming will be:

&&USER-ID.REQ**&&REQ-NUMBER**

3.1.2. UNIX & Windows Natural Server Platforms

For UNIX (and variants) or Windows Natural servers, the default naming will be:

3.1.2.1. JCL / Script Reference

The default naming for a **Request File JCL / Script Reference** will be:

&&USER-ID&&REQ-NUMBER

3.1.2.2. FTP Reference

The default naming for a **Request File FTP Reference** will be:

&&USER-ID&&REQ-NUMBER

3.2. Output File(s) tab

Output File(s) are the names of the files that NatQuery requests will write data into on the server platform. Each NatQuery user will have at least one **User Output File** specified, with each defined **User Output File** being "paired" with a corresponding **Request File** (except in the case of a Mainframe Natural server and the use of Direct FTP as mentioned above where Request Files are not needed).

3.2.1. Mainframe Natural Server Platforms

The default naming for **Output Files** against mainframe Natural servers will be:

3.2.1.1.JCL / Script Reference

The default naming for an **Output File JCL / Script Reference** will be:

&&USER-ID.OUT**&&REQ-NUMBER**

3.2.1.2.FTP Reference

the default naming for the **Output File FTP Reference** will vary depending on the type of mainframe server.

For **MVS** Systems when the **FTP Session Establishes Working Directory** option is set, the default naming will be:

OUT**&&REQ-NUMBER**

In all other situations, the default naming will be:

&&USER-ID.OUT**&&REQ-NUMBER**

3.2.2. UNIX & Windows Natural Server Platforms

For UNIX (and variants) or Windows Natural servers, the default naming will be:

3.2.2.1.JCL / Script Reference

The default naming for a **Output File JCL / Script Reference** will be:

&&USER-ID&&REQ-NUMBER _OUT.TXT

3.2.2.2. FTP Reference

The default naming for a **Output File FTP Reference** will be:

&&USER-ID&&REQ-NUMBER _OUT.TXT

3.3. Log File tab

Log Files are used to capture the name of the file that will record the execution status of a user's submitted request on the Natural server platform. Each NatQuery user will typically have only a single **Log File**.

3.3.1. Mainframe Natural Server Platforms

The default naming for **Output Files** against mainframe Natural servers will be:

3.3.1.1. JCL / Script Reference

&&USER-ID.LOGFILE

3.3.1.2. FTP Reference

the default naming for the **Log File FTP Reference** will vary depending on the type of mainframe server.

For **MVS** Systems when the **FTP Session Establishes Working Directory** option is set, the default naming will be:

LOGFILE

In all other situations, the default naming will be:

&&USER-ID.LOGFILE

3.3.2. UNIX & Windows Natural Server Platforms

For UNIX (and variants) or Windows Natural servers, the default naming will be:

3.3.2.1. JCL / Script Reference

&&USER-ID_LOG.TXT

3.3.2.2. FTP Reference

&&USER-ID_LOG.TXT

4. Constraints frame

The **Constraints** frame captures a single piece of information that indicates whether or not a user, by default, will be defined as being constrained to being **Execute-Only**; I.E. the user would not be allowed to create new extracts requests, they will only be able to execute existing queries.

It is suggested to leave the **Execute-Only** checkbox un-checked.

5. Data Access Limits frame

The **Data Access Limits** frame captures the abilities and limits that may be imposed on a user.

5.1. Allow Read Physical

The **Allow Read Physical** checkbox indicates whether or not the user will be allowed to create an extract that does a Read Physical of a file. By default this will be selected, and the person performing the install may wish to reflect on this setting in the context of whether or not it is appropriate for new users who might be added later to NatQuery. The value selected will be used as the default value for a new user definition, this value can be over-ridden when each individual user is defined.

Determine what the default value for new users should be and adjust accordingly.

5.2. Allow Read By ISN

In specific situations, it is desirable to allow a user to perform a Read by ISN – for example to insure that no redundant records are returned by a query that must read all records in a file.

By default, this option will not be selected.

Determine what the default value for each new user should be and adjust accordingly.

5.3. Limit Record

The **Limit Records** option allows for the placing of a limit on how many records a given user may read by default from the outermost I/O loop that will be generated by NatQuery. If the Limit option is selected, then NatQuery will allow the entry of a Limit Number in a text box.

If a **Limit Records** is specified and a Limit number is provided, then a NatQuery–generated extract will terminate once this threshold has been met.

Determine if a Limit other then the default should be applied to each new, and if so; what the limit should be.

5.4. Accounting frame

The **Accounting frame** allows for the entry of billing information that may be optionally dynamically substituted into generated JCL / Script for subsequent capture by chargeback systems. If there is no job accounting, then the text box labeled **Major Account**, as well as the text box labeled **Sub-Account** may both be left blank. If there is a job accounting prerequisite then these text boxes should be filled in appropriately.

By default the Accounting frame text boxes will be blank, for each new, user determine if the default values apply.

Step 3.2 – Create Initial User Profile

With the above steps followed, the **Administer User Information** window may now be closed by clicking the **OK** button. This action will return the user to the **Administer Users** window, which should now show a **Default** user as being defined.

Click the **New** button to create a new user; this will invoke the **Administer User Information** window, with defaults provided from the **Default User Profile** just built. Perform the following:

1. **Enter User-ID**
   Into the text box labeled **User ID**; enter the User ID that was provided as the **Server User ID** in Step 1, being careful to exactly match the nuances of upper and lower case (if any).

   Note:
   While the **User-ID** value is being entered, the information displayed in the **File References** frame will be automatically adjusted with the appropriate substitution(s) for **&&USER-ID** and **&&REQ-NUMBER** based on how these values may be referenced in the Default User profile.

2. **Review Active Requests**
   In the upper right-hand corner of the **Administer User Information** window is a text box labeled "**Active Requests**". If the default was used when the Default User profile was defined as outlined in preceding steps, then the number shown in **Active Requests** should show as 3.

   The number shown in the **Active Requests** text box controls how many current requests (or "Request Slots") the user will be allowed to have. While the default of 3 is an acceptable number for general users of NatQuery, for Administrators it is often very useful that this number be increased. This allows the Administrator greater flexibility when handling the tasks that will allow NatQuery to be configured (tasks which will utilize these Request Slots.

   For Administrative users then, *it is suggested* that the number of **Active Requests** be increased to 9.

3. **Enter User Name**
   Into the text box labeled **User Name**; enter the name of the user associated with the **User-ID** just entered.

4. **Enter User-Email**
   If desired; enter the e-mail address associated with the **User-ID** just entered. This information is optional.

5. **Review User File References**
   With a **Default User Profile** created, the user just added will inherit the attributes of the **Default User Profile**. As these instructions assume that the user being added is in fact

also a NatQuery Administrator, **File References** should now be reviewed, with any necessary adjustments to the default profile being made.

As the name of the **JCL / Script Reference** for the user's **Log File** will be needed in a subsequent step, it is a good idea to make a note of the name of this file now. If the above instructions have been followed, the name of this file will be one of the following:

MVS and VSE servers:

&&USER-ID.LOGFILE

UNIX, Linux, Windows and OpenVMS servers:

&&USER-ID_LOG.TXT

where **&&USER-ID** is the value entered into step 1 above.

6. **Review Remaining Constraints & Data Access Limits**
   With a Default Profile created, the user just added will inherit the attributes of the default profile. As these instructions assume that the user being added is in fact also a NatQuery Administrator, the default values found in the **Constraints** and **Data Access Limits** should now be reviewed, with any necessary adjustments to the default profile being made.

7. **Click OK to close the Administer User Information Window**
   The user should now click the **OK** button to close the **Administer User Information** window; this action will return the user to the **Administer User** window, which will now show the Default user and the newly added user.

   While any additional users of NatQuery can be defined at this time, it is suggested that the Administrator continue to focus on completing the building of an Environment Configuration, such that new users are added later.

8. **Click OK to close the Administer Users Window**
   The user should now click the **OK** button to close the **Administer Users.**

   As the **Administer Users** window closes, NatQuery will prompt the user with an Verify Question message similar to the one seen if Figure 3 (**Verification Question**).

   Since the process of building an Environment Configuration is still underway and numerous deficiencies still exist that will be addressed in subsequent steps, it is suggested that the user respond with **No** to this prompt (thus bypassing the verification of the current Environment Configuration).

   NatQuery will then respond with message similar to that seen in Figure 2 (**Un-Verified Environment**). This message is normal at this point, and can be ignored by clicking the

OK button.

The user will then be returned to the NatQuery desktop and can continue with the next section entitled Step 4 - Create Initial JCL / Script Templates.

Step 4 - Create Initial JCL / Script Templates

The configuration process will now focus on handling the JCL / Script templates that NatQuery will use to initially interact with the Natural server platform.

While NatQuery supports several different types of JCL / Script templates, the JCL / Script templates that are minimally required will vary depending upon the intended use of NatQuery.

To automate the processes of obtaining Data Definition Modules (DDMs), a JCL / Script template called **Production Special Process** would be required. However, since the process required to obtain the needed DDMs can also be run completely manually (with NatQuery then being capable of Importing the DDM information), the **Production Special Process** template can be considered optional although *it is recommended* that this template be built.

To automate several Administrative functions and also to utilize typical End-User Data Extraction requests, NatQuery has a requirement that the JCL / Script template called **Production Request Process** be built.

To automate Data Warehouse extraction, NatQuery has a requirement that the JCL / Script template called **Production DWH Process** be built.

Prior to proceeding with this step, it should be noted that the goal of this step will ***NOT*** be to provide fully-functional JCL / Script templates. Rather, the goal will be to provide NatQuery with the core JCL / Script templates that may be required by utilizing installation-provided example JCL / Script templates, with only minor regard (at this point) as to whether or not these templates will actually function as expected. By taking this approach, configuration can proceed by fulfilling the requirements that one or more of these 3 templates exist, and the work required to insure that these JCL / Script templates are fully functional in the subsequent configuration steps. In this way, the debugging of any issues that may exist with the execution of JCL / Script templates can be debugged within the context of a specific processing task that attempts to make use of these templates.

Perform the following steps:

1. **Invoke Administer JCL / Script Function**
   To invoke the Administer JCL / Script function, the user will start on an empty NatQuery desktop, and then click the **Administer** drop-down menu, then click on **Environment Configuration / Server Connection Configuration / JCL/Script Information**. This will invoke the **Administer JCL / Script** function.

2. **Build Production Request Process Template**
   the **Production Request Process** template is a template that is required to exist for all configurations of NatQuery.

   At the top right of the **Administer JCL / Script** window is a combo box named **Server JCL**

/ Script Component.

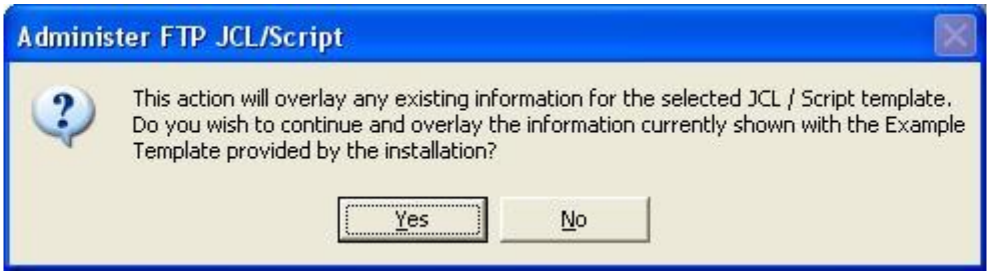
2.1. Insure that **Server JCL / Script Component** has the value **Production Request Process** selected; select this value if it is not selected.

In most cases this template will not exist, and therefore the text box labeled **Production Request Process JCL / Script Template** will show a single line of text:

"This template does not contain any JCL / Script Information"

2.2. Click the button named **Copy From Example Template**, found in the top right-hand corner of the window.

This action will result in the presentation of a message box similar to the following:


Administer FTP JCL/Script

This action will overlay any existing information for the selected JCL / Script template. Do you wish to continue and overlay the information currently shown with the Example Template provided by the installation?

Figure 4 – JCL Overlay Message

This message is normal, and the user should click the **Yes** button.

2.3. The Administrator should now examine the JCL template for any obvious changes that may need to be made in order for this JCL to execute as provided. Of particular note would be such things as any required JOB Cards (applicable to Mainframe Natural servers), references to disk usage (applicable to Mainframe Natural servers), and how Natural itself is invoked (applicable against all Natural Server platforms).

When performing this step, it is highly desirable to have a working example of a Natural Batch JCL / Script stream to use as a reference to the types of changes that may be required.

For detailed information on the proper construction of JCL / Script templates, in particular the **Production Request Process** template, the user may wish to refer to the chapter entitled JCL / Script Template.

2.4. When the JCL appears to be correct, the Administrator should click the **Save** button. This action will save the **Production Request Process** template as a file into the current Environment Configuration path.

3. **Build the Production Special Process Template**
The **Production Special Process** template is a template that is recommended for all

configurations of NatQuery (although this template is not absolutely required).

At the top right of the **Administer JCL / Script** window is a combo box named **Server JCL / Script Component**.

3.1. Set the **Server JCL / Script Component** combo box so that **Production Special Process** is selected.

In most cases this template will not previously exist, and therefore the text box labeled **Production Special Process JCL / Script Template** will show a single line of text:

> "This template does not contain any JCL / Script Information"

3.2. Click the button named **Copy From Example Template**, found in the top right-hand corner of the window.

This action will result in the presentation of a JCL Overlay message similar to that seen in Figure 4 (**JCL Overlay Message**). This message is normal, and the user should click the **Yes** button.

3.3. The Administrator should now examine the JCL template for any obvious changes that may need to be made in order for this JCL to execute as delivered. Of particular note would be such things as any required JOB Cards (applicable to Mainframe Natural servers), references to disk usage (applicable to Mainframe Natural servers), and how Natural itself is invoked (applicable against all Natural Server platforms).

When performing this step, it is highly desirable to have a working example of a Natural Batch SYSTRANS / SYSOBJH JCL stream to use as a reference to the types of changes that may be required.

For detailed information on the proper construction of JCL / Script templates, in particular the **Production Special Process** template, the user may wish to refer to the chapter entitled JCL / Script Template.

3.4. When the JCL appears to be correct, the Administrator should click the **Save** button. This action will save the **Production Special Process** template into the current Environment Configuration.

4. **Build the Production DWH Process Template**
The **Production DWH Process** template is only required if one of the primary intentions of the use of NatQuery is to serve as a data extraction engine for integration of ADABAS data into ETL tools. If integration to ETL tools is not an intention, then this step may be bypassed.

At the top right of the **Administer JCL / Script** window is a combo box named **Server JCL**

/ Script Component.

4.1. Set the **Server JCL / Script Component** combo box so that the value **Production DWH Process** selected.

In most cases this template will not previously exist, and therefore the text box labeled **Production DWH Process JCL / Script Template** will show a single line of text:

> "This template does not contain any JCL / Script Information"

4.2. Click the button named **Copy From Example Template**, found in the top right-hand corner of the window.

This action will result in a message similar to that seen in Figure 4 (**JCL Overlay Message**). This message is normal, and the user should click the **Yes** button.

4.3. The Administrator should now examine the JCL template for any obvious changes that may need to be made in order for this JCL to execute as designed. Of particular note would be such things as any required JOB Cards (applicable to Mainframe Natural servers), references to disk usage (applicable to Mainframe Natural servers), and how Natural itself is invoked (applicable against all Natural Server platforms).

When performing this step, it is highly desirable to have a working example of a Natural Batch SYSTRANS / SYSOBJH JCL stream to use as a reference to the types of changes that may be required.

For detailed information on the proper construction of JCL / Script templates, in particular the **Production DWH Process** template, the user may wish to refer to the chapter entitled JCL / Script Template.

4.4. When the JCL appears to be correct, the Administrator should click the **Save** button. This action will save the **Production DWH Process** template into the current Environment Configuration.

5. **Close Administer JCL / Script Function**
The user may now close the Administer JCL / Script window by clicking the **OK** button.

As this window closes, a message box will be displayed similar to the one seen in Figure 3 (**Verification Question**).

At this point, it is suggested that the user click **Yes** to this prompt; this will invoke a **Verify Configuration** window with this window displaying the general status of the current Environment Configuration being built.

The most salient note in the **Verify Configuration** will be found at the absolute bottom of the report. If the above instructions were followed, and if FTP was selected as the **Server**

Communication Mode: The report should indicate that **FTP mode is verified for use**. Otherwise, if PC Network was selected as the Server Communication Mode, the report should indicate that **PC Network is verified for use**.

If the user desires to better understand what a Verify Configuration accomplishes, the user should refer to the chapter entitled Verifying an Environment Configuration.

The Administrator may now continue with the next section entitled Step 5 - Server Platform Initialization.

Step 5 - Server Platform Initialization

The configuration process will now focus on work that is required on the Natural server platform.

In order to operate as designed, NatQuery will depend on interaction with one of several Natural programs that must reside in Natural on the server platform, and it will further depend on the interaction of files that must exist on the server platform.

The actions required to handle the **Server Platform Initialization** will be dependent upon the type of platform that Natural resides on.

For sites that are using NatQuery against a mainframe Natural server platform, the user should follow the steps outlined in Step 5.1 - Mainframe Server Initialization.

For sites using NatQuery against UNIX, Linux and Windows, then the user should follow the steps outlined in Step 5.2 - Open Systems Server Initialization.

Step 5.1 - Mainframe Server Initialization

The steps in this section should only be performed if the Natural Server platform resides on a mainframe. In all other situations, the section entitled NatQuery Open Systems Server Initialization should be followed.

Perform the following steps:

1. **Define VSAM catalog (VSE Systems using VSAM ONLY)**
   If you are installing NatQuery to operate against an MVS system, the user should proceed to the next step. This step should only be performed when NatQuery is used against a VSE system.

   In order to operate as expected against VSE systems, NatWorks suggests the use of a VSAM user catalog to contain all required NatQuery-related datasets. The reason for using VSAM (or more precisely VSAM for SAM) for this purpose is that VSAM allows for files to be allocated and defined with additional extent information, such that extracts will not be as susceptible to abending due to exceeding an estimated size (as might occur by using straight sequential files).

   As an alternative to using VSAM, a site may also use a facility such as DYNAM, in which case this step can be ignored and the user should move to the next step.

   If the use of VSAM is desired, then through the use of IDCAMS or a similar utility, the user should allocate and define a VSAM user catalog that will be used to contain all NatQuery related mainframe files. The amount of space required for this allocation will vary from site to site depending upon several factors, including the number of NatQuery users and the estimated size of the largest extracts. At a minimum, it is suggested that several hundred cylinders be allocated to this catalog.

   For example purposes, the physical name of this catalog is suggested to be "NATWORKS.CATALOG", with this catalog being referenced under the name of "NATWORK".

2. **Define User Log File (Both MVS and VSE Systems)**
   For both MVS and VSE systems, a disk dataset should now be manually defined that will become the **User Log File** of the user defined in step 1 of the section **Create Initial User Profile**.

   For MVS systems, this can be accomplished through the use of ISPF or a similar system utility.

   For VSE systems, this can be accomplished either through the use of IDCAMS or similar system utility, with the allocation of this file being placed into the VSAM catalog defined in the previous step, or alternative defined in DYNAM or similar storage facility.

Allocate the user's **User Log File** dataset so that it has the following definition:

2.1. The dataset name should match the JCL Reference of the initial user's Log File as reviewed in step 4 of the section **Create Initial User Profile** above.

For MVS systems, *it is recommended* that this file be named as:

"***userid***.NATQUERY.LOGFILE

where "***userid***" is the value of the User ID specified in step 1 of the section **Create Initial User Profile** above.

For VSE systems when VSAM is being used, then *it is recommended* that this file be named as:

"***userid***.LOGFILE"

where "***userid***" is the value of the User ID specified in step 1 of the section **Create Initial User Profile** above. When this file is defined to the VSAM catalog, it should be defined as a SAM ESDS sequential file. When defined, this file should use the IDCAMS parameters of NONINDEXED and REUSE.

For VSE systems when another storage facility such as DYNAM is being used, *it is recommended* that this file be named as:

"***userid***.NATQUERY.LOGFILE

where "***userid***" is the value of the User ID specified in step 1 of the section **Create Initial User Profile** above.

2.2. The dataset should be allocated as FB, with an LRECL of 80, an appropriate Blocksize (8000 is suggested), and at least 1 block of disk space.

3. **Modify TCP/IP Startup JCL (VSE Systems using VSAM ONLY)**
If you are installing NatQuery to operate against an MVS system, the user should proceed to the next step. This step should only be performed when NatQuery is used against a VSE system and a VSAM catalog is being used.

In order to operate as expected against VSE systems, TCP/IP and FTP functionality is typically provided through software developed by a company called Connectivity Systems.

In order to instruct FTP to treat the VSAM catalog as a directory, the following (or similar) statement must be included in this TCP/IP startup stream, under the File System portion of this stream:

```
DEFINE FILE,DLBL=NATWORK,PUBLIC='NATWORK',TYPE=VSAMCAT
```

To allow access to this VSAM catalog, an appropriate DLBL in either standard labels, partition labels or temporary labels must be provided, similar to the following:

```
// DLBL NATWORK,'NATWORKS.CATALOG',,VSAM,,CAT=NATWORK
```

While reviewing this stream, it should be determined whether or not FTP has been defined to operate against the VSE POWER queue. This definition will be critical in NatQuery's ability to use Direct FTP, as it is this definition that allows FTP to access POWER. The definition of POWER to FTP will be similar to the following:

```
DEFINE FILE,PUBLIC='POWER',DLBL=IJQFILE,TYPE=POWER
```

4. **Upload NatQuery Server Programs (both MVS and VSE Systems)**
   In order to operate as intended against either an MVS or a VSE system, NatQuery will depend on the use of several Natural programs that will handle various aspects of user Log File handling on the server platform.

   For mainframe environments, there are three Natural programs (NQYP0002, NQYP0003 and NQYP0004) that need to be introduced into Natural, and the **Upload NatQuery Server Programs** function of NatQuery allows for the automated introduction of these Natural programs into Natural on the mainframe.

   To invoke the **Upload NatQuery Server Programs** function, the user will have NatQuery started with NatQuery showing an empty desktop.   The user will then click on **Administer / Environment Configuration / Server Connection Information / Upload NatQuery Server Programs**.

   This action causes NatQuery to generate a JCL file based upon the **Production Request Process** JCL template, with this JCL file being built so that it should introduce the required Natural programs into mainframe Natural.

   Once the JCL file has been generated, and assuming that NatQuery has not yet performed an FTP operation within the same NatQuery session, then NatQuery will bring up a **FTP Password** window that prompts for the user's FTP password with a message box similar to the following:

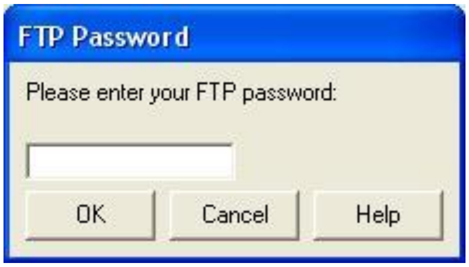
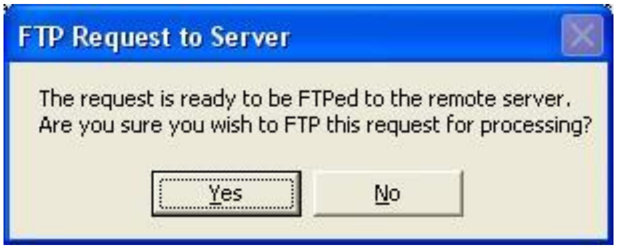


Figure 5 – FTP Password Question

In response to this Input box, the user should provide the appropriate FTP Password associated with User ID provided to NatQuery, and then click the **OK** button on the **FTP Password** message box.

If NatQuery has previously successfully performed an FTP operation with the same NatQuery session (which would not usually be the case at this point in a new configuration), or the user just provided the appropriate FTP password to the preceding **FTP Password** prompt, then the user will be prompted for confirmation of the FTP operation with a message box similar to the following:


FTP Request to Server

The request is ready to be FTPed to the remote server.
Are you sure you wish to FTP this request for processing?

Figure 6 – FTP Request to Server

In response to this prompt, the Administrator should click the **Yes** button in response.

If NatQuery has been configured to use **Direct FTP**, then a generated JCL stream should have been placed directly into JES (MVS) or POWER (VSE). If this job was released into JES or POWER and the job has not yet executed automatically, then the job should be manipulated so that it does execute, or the Administrator should wait until it has executed before continuing.

If NatQuery has been configured to use **Just FTP**, then a generated JCL stream should have been placed onto the mainframe server as a file. How the file will be named will be dependent on several factors. If the suggested defaults were accepted when the Default User Profile was handled, then at the lowest level the file will be named similar to the following

&&USER-ID.REQ**&&REQ-NUMBER**

where **&&USER-ID** is equal to the value of the User ID that NatQuery is operating under, and &&REQ-NUMBER is the request slot that the request was submitted into (usually “1”, “2” or “3” if defaults were followed). In some cases (such as when a VSAM catalog is being used, FTP session don’t establish a working directory, or User File Directories are referenced) this naming convention may have an addition high-qualifier, in other situations (such as when other naming conventions are employed) the file may be named completely differently. Once the file is located, the Administrator should take whatever steps are necessary so that the generated JCL is submitted for execution in batch and that execution has completed before continuing. If the Administrator has difficulty in locating the file, then the Troubleshooting section should be referenced.

If NatQuery encounters an error with the FTP operation to move the generated JCL onto the mainframe, then the Troubleshooting section of this manual should be referred to. Once corrective action has been taken to the template, this step should be re-done.

If the visual examination of the execution results reveals errors, then in almost all cases these will generally have something to do with errors in how the JCL was generated. To correct such issues, the Administrator should make whatever changes are necessary to the **Production Request Process** template, and this step should then be re-done. If the administrators have questions on how the **Production Request Process** template should operate, or how this template needs to be changed, then the user should refer to the section of the manual entitled JCL / Script Template.

If problems are seen with the execution of the JCL, then the problem is usually resolved by making changes to the Production Request Process template
At this point, it is assumed that the JCL has been executed – and the Administrator is advised to visually check into the execution status of the executed JCL to look for any difficulties that are likely to exist. In all cases, if any problems are found, these issues are likely to be resolved by making changes to the **Production Request Process** template.

If questions on how the **Production Request Process** template should operate arise, then the user should refer to the section of the manual entitled JCL / Script Template section.

If issues arise with the execution, then the Troubleshooting section should be referred to.

When the execution of the JCL / Script indicates a successful execution, then the Natural library designated in step 1 of the section **Natural Server Information** should be reviewed for the existence of both source and object code for the programs NQYP0002, NQYP0003 and NQYP0004.

As an alternative to using NatQuery to introduce the required programs using the **Upload NatQuery Server Programs** function, it is possible to manually handle the loading of the required Natural programs. In this case, the user may utilize either the SYSTRANS or SYSOBJH utilities of Natural to accomplish this using as input:

SYSTRANS_MAINFRAME_SYSTEMS.TXT (for use with SYSTRANS), or

SYSOBJH_MAINFRAME_SYSTEMS.TXT (for use with SYSOBJH)

These files will be located in the FILES sub-directory of the NatQuery install directory (usually C:\Program Files\NatQuery).

Once the required programs have been introduced into the appropriate library of Natural on the server platform, the user can continue with the next section entitled **Environment Configuration**.

Step 5.2 - Open Systems Server Initialization

The steps in this section should only be performed if the Natural Server platform resides on an Open System platform such as Windows, UNIX or Linux. Against all other platforms (I.E. mainframes), the preceding section entitled **NatQuery Mainframe Server Initialization** should be followed.

Perform the following steps:

1. **Define User Log File**
   For all Open System platforms, a disk dataset should now be manually defined that will become the **User Log File** of the user defined in step 1 of the section **Create Initial User Profile**.

   Using a text editor against the server platform, create a text file that contains one space character / one blank line. Save this file as ***userid***_LOG.TXT (this name was the suggested value to be used when the Administrator's User ID was defined to NatQuery in step 4 of the section **Create Initial User Profile)**.

2. **FTP Upload NatQuery Server Programs**
   In order to operate as intended against an Open System platform, NatQuery will depend on the use of several Natural programs that will handle various aspects of logging on the server platform.

   For Open Systems environments such as UNIX, Linux or Windows, there are three programs that need to be introduced into Natural (NQYPNT03, NQYPNT05 and NQYPNT06). The **Upload NatQuery Server Programs** function of NatQuery allows for the generation of the files that will accomplish this introduction using the SYSOBJH utility of Natural, NatQuery will be configured to move these files directly onto the server platform. How these files will then be executed will be dependent on the selected method of **Automated Execution** as described below.

   To invoke the **Upload NatQuery Server Programs** function, the user will have NatQuery started with NatQuery showing an empty desktop. The user will then click on the **Administer / Environment Configuration / Server Connection Information / Upload NatQuery Server Programs**.

   This action causes NatQuery to build a base Script file based upon the **Production Special Process** Script template (as well as several other files that will support the execution of the Script file). When the Script is executed, it will attempt to introduce the required NatQuery server programs to the remote Natural environment on the mainframe and into the Natural Library identified in step 1 of the section **Natural Server Information**.

   If NatQuery is configured to use **FTP**, and NatQuery has not yet performed an **FTP** operation previously within the same NatQuery session, then NatQuery will respond to this action by prompting the user for the user's FTP password. If this is the case, then the user should provide the appropriate FTP password and then click the **OK** button on the **FTP Password** message box.

If NatQuery is configured to use **FTP** and NatQuery has previously successfully performed an **FTP** operation within the same NatQuery session, or the user just provided the appropriate FTP password to the preceding prompt, then the user will be prompted for confirmation of the FTP operation, and the user should click the **Yes** button.

If NatQuery is configured to use **PC Network**, the user will be prompted for confirmation of the PC Network Copy operation, and the user should click the **Yes** button.

Depending upon the method of **Automated Execution** selected (I.E. **Direct FTP** or **Direct Copy** versus **Just FTP** or **Just Copy** – processing that requires Remote Execution capabilities) then NatQuery may now prompt the user for confirmation as to whether or not Remote Execution will be used.

This step will allow you to both test the **FTP** or **PC Network** Copy capabilities of NatQuery, as well as the construction of the **Production Special Process** JCL template.

If NatQuery encounters errors with either **FTP** or **PC Network** Copy operations, then the Troubleshooting section of this manual should be referred to, and once corrective action has been taken this step should be re-run.

If NatQuery encounters errors with Script execution, then the **JCL / Script Templates** section of this manual should be referred to, and once corrective action has been taken this step should be re-run.

If the execution of the Script indicates a successful execution, then the Natural library designated in step 1 of the section Natural Server Information should be reviewed for the existence of both source and object code for the programs NQYPNT03, NQYPNT05 and NQYPNT06. Once these programs have been introduced, the user can continue with the next section entitled **Environment Configuration**.

As an alternative to using NatQuery to introduce the required programs, it is possible to manually handle the loading of the required Natural programs. In this case, the user may utilize either the SYSTRANS or SYSOBJH utilities of Natural, using as input:

SYSTRANS_OPEN_SYSTEMS.TXT (for use with SYSTRANS), or

SYSOBJH_OPEN_SYSTEMS.TXT (for use with SYSOBJH)

These files will be located in the FILES sub-directory of the NatQuery install directory (usually C:\Program Files\NatQuery).

3. **Create NatQuery Specific NATPARM File**

To separate any parameters that may require to be set for NatQuery / NatCDC executions, it is recommended that the Administrator create a copy of the current Natural Batch parameter file, or a copy of the NATPARM file (if a Batch-specific parameter file does not exist) and save this as a Natural Parameter File named **NATQUERY**.

Doing this accomplishes two things. First, it allows for parameters that may be specific to NatQuery to be changed without impacting any other users. Secondly (and perhaps more importantly), the Script templates that are provided by a NatQuery install will expect to use a Parameter file called **NATQUERY**.

4. **Copy USR Modules Into Natural Library**
   In order to function as designed on Open System, NatQuery currently makes use of two User Exit Modules (USR Modules), USR0080N and USR1052N.

   The User Exit Module USR0080N handles the introduction of a NatQuery-generated program into Natural, with USR0080N being called by the NatWorks provided program NQYPNT05 – which perform the introduction of an external file into Natural as a program.

   The User Exit Module USR1052N handles the ability to issue a command to the Operating System (I.E. a Shell command), with most NatQuery generated programs utilizing this call to perform a cat command (to handle user log files).

   Both USR0080N and USR1052N will need to be copied from the Software AG supplied library called SYSEXT into the Natural target library defined to NatQuery, with this Copy being handled by the Natural utility SYSMAIN.

The configuration process can now continue with the next section entitled **Environment Configuration**.

Step 6 - Environment Configuration

In order to provide customer-specific data processing, NatQuery must be supplied with information that pertains to the files (usually ADABAS files) that NatQuery will be expected to handle. With the Natural Programming Language this information is encapsulated into objects called Data Definition Modules (DDMs) and because of this fact, DDMs are the basic building blocks of NatQuery's generation engine.

The process of building an **Environment Configuration** therefore begins by providing NatQuery with one or more DDMs. Once one or more DDMs are made available to NatQuery, the Administrator will then proceed to provide additional information that relates to these DDMs.

The process of handling DDMs with NatQuery is detailed in the section Adding a New DDM, and the Administrator should now refer to this section.

As a result of successfully completing the above steps, there should now be a fully configured Environment Configuration present in the path designated by the current value of **Environment Path** (as specified through the **Environment Path** in the **Environment Paths** tab of the **NatQuery Configuration** function).

With this **Environment Configuration** established, the workstation upon which the Environment Configuration was built can now be immediately utilized to generate Natural data extraction programs.

To allow for this **Environment Configuration** to be usable to other NatQuery workstations in a networked environment, the **Environment Configuration** must be made available to these workstations.

For further information on how to provide the **Environment Configuration** to other NatQuery installations, please refer to the section entitled **Rolling Out an Environment Configuration**.

Environment Configuration

A Full Environment Configuration is a collaboration of a several different collections of information which, when combined, allow NatQuery to function as intended.

Prior to performing any work on an existing Environment Configuration, it is desirable that an Administrator first review the section entitled Establishing a Working Environment so that potential negative impact to existing users will not occur.

General aspects of an Environment Configuration would be:

- **User Maintenance**
  To perform User Maintenance, please refer to the following section entitled User Maintenance.

- **DDM Maintenance**
  To perform DDM Maintenance, please refer to the following section entitled DDM Maintenance.

- **Global Variable Maintenance**
  To perform Global Variable maintenance, please refer to the section entitled Global Variables.

- **JCL / Script Maintenance**
  To perform JCL / Script Maintenance, please refer to the JCL / Script Template section of this manual.

- **Natural Server Information Maintenance**
  To perform maintenance to information relating to Natural and how NatQuery interacts with Natural, please refer to the section entitled Natural Server Maintenance.

- **Connectivity Information Maintenance**
  To perform maintenance to information relating to how NatQuery connects to and interact with the Natural server platform, please refer to the section entitled Server Connection Maintenance.

- **Verifying an Environment Configuration**
  Information relating to the NatQuery Verification process can be found in the section entitled Verifying an Environment Configuration.

- **E-mail Enabling of Output**
  Information relating to allowing NatQuery extracts to automatically e-mail extracted output are described in the section entitled Enabling E-Mail handling of Extracted Output.

Note 1:
In order to be able to perform any modification of an **Environment Configuration**, an Administrator version of NatQuery (I.E. a version that has been supplied with an Administrator License Key) must be used.

Note 2:
When changes are required to be made to a working "Production" Environment Configuration, the Administrator ***must*** be aware that the process of making changes to a Configuration can result in a given Configuration becoming temporarily disabled, either partially or totally, until such time as the Environment is verified for use through the execution of the Verify Environment Configuration function, and the various levels of verification are passed.

Prior to making any changes to an Environment Configuration, it is strongly suggested that a backup of the current Environment Configuration files be made. To make an external backup of the current Environment Configuration, copy the entire contents of the path designated on the NatQuery toolbar as the **Environment Path**.

Beyond making a backup, it is strongly suggested that an additional copy of the existing Environment Configuration be placed into a "working" directory, with the Administrator's installation of NatQuery then being configured to have its Environment Path resolve to this working directory. Required changes can then be made against this working directory at the discretion of the Administrator without impacting any current users. Once all required changes are made to the working directory and the contents of the working directory passes Verification to the Administrator's satisfaction, the Administrator can then Export from the working directory into the production directory.

To establish a working directory for an existing Environment Configuration, please refer to the section entitled Establishing a Working Environment.

Establishing a Working Environment

This procedure is only necessary when the Administrator desires to make changes to an existing NatQuery Environment Configuration, or the Administrator would otherwise like to preserve a currently working Environment Configuration. This procedure is not necessarily required in situations where there is only one installation of NatQuery, or where there are multiple installations but all of these installations point at Environment Configurations that are local to each machine (I.E. non-networked).

To establish a Temporary Environment Configuration, perform the following:

1. Start NatQuery if it is not already started. Details on how to start NatQuery can be found in the section entitled Starting NatQuery.

2. On the NatQuery desktop (and with no query open), click on the **Administer** drop-down menu and then click the **NatQuery Configuration** option. This will invoke the **NatQuery Configuration** window.

3. On the **NatQuery Configuration** window, click on the **Environment Paths** tab.

4. The enterable text fields on the **Environment Paths** tab instructs NatQuery as to where it should either find or place various Environment Configuration files that describe the FTP and ADABAS File environment. At this point you should make note of the current setting of **Environment Path**, as this path controls where NatQuery is currently resolving its Environment Configuration from. Assuming that the current designation of the Environment Path actually contains the Environment Configuration in which changes are required, you will need to reference this directory in a subsequent step. If the current designation of **Environment Path** does not contain the directory of the current Environment Configuration, then the Administrator must know what this path is.

   At this point you will either create or select a “working directory” that will serve to contain a copy of the current Environment Configuration that you intend to modify.

   If the directory that will be used to contain the “working directory” does not exist, then this directory will need to be created. This can be accomplished by clicking on the **Create Folder** button located at the bottom of the **Environment Paths** tab. Clicking on the **Create Folder** button will invoke the **Create New Folder** window that will allow you to create the desired directory. Help on using the Create New Folder window is available by clicking on the **Help** button located on this window.

   At this point, either a new directory was just created that will serve as a “working directory”, or a suitable directory already existed. The value of this “working directory” should now be given to NatQuery as the value of the Environment Path. This can be accomplished manually by entering the full path to the “working directory” into the **Environment Path** text box, or can be selected in an automated fashion by using the **Browse** button located to the right of the Environment Path textbox.

After the full path of the "working directory" has been placed into the text field associated with **Environment Path**, click the **OK** button. Depending upon whether there is anything in the newly selected "working directory", NatQuery may respond with an initialization message indicating that the current Environment Path setting is invalid or non-existent. If NatQuery produces this message, this message can be ignored.

5. On the NatQuery desktop, click on the **Administer** drop-down menu. Click on the **Environment Configuration** menu item, and then click on the **Import Environment Configuration** menu item. This will invoke the **Import Environment Configuration** window.

6. Using the controls on the **Import Environment Configuration** window, import the current Environment Configuration from the directory that contains this into the designated "working directory". Specific help on using this function can be found by clicking on the **Help** button while in this function. Once the import is complete, click the **OK** button to close the **Import Environment Configuration** window.

7. With the current Environment Configuration imported into the "working directory", you are now free to perform maintenance against this Environment Configuration in isolation from any users. Instructions on the common types of modifications are found later in this section. Once whatever steps are required for maintenance are performed, the following steps should then be executed to complete the modifications.

8. Once all modifications have been made to the Environment Configuration in the "working directory", the Administrator should verify the modified Environment Configuration to insure its validity. This verification is performed through the **Verify Environment Configuration**, and is executed by first clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration**, and then clicking on **Verify Environment Configuration**. Specific help on using this function is available by clicking on the **Help** button while in this function.

   The **Verify Environment Configuration** function produces a report that pertains to the information found in the currently specified Environment Configuration. This report should be examined in detail to insure that all aspects of the Environment Configuration are accurate and complete. If unacceptable errors are found, then the Administrator should correct these errors using the appropriate Administration functions of NatQuery (Administer File Relationships, Administer Descriptor Statistics, Administer Occurrence Information) until executing the **Verify Environment Configuration** report matches the Administrator's expectations. If no unacceptable errors are found, then the next step can be executed.

9. Once the Environment Configuration in the currently specified "working directory" is verified to the Administrator's satisfaction, the Administrator can then "roll-out" the Environment Configuration as required. For detailed information on how to perform this "roll-out", please refer to the section entitled Rolling Out and Environment

Configuration.

10. As a final step (and particularly suitable when NatQuery is installed so that End-Users share a network directory that contains the Environment Configuration), the Administrator may now desire to change the current setting of Environment Path setting on their machine back to the to the correct network directory. This is accomplished through the **NatQuery Configuration** function, with this function being available from the **Administer** drop-down menu. Once on the **NatQuery Configuration** function, the **Environment Paths** tab contains the field **Environment Path** – with the setting of **Environment Path** designating to NatQuery the path where the Environment Configuration information will be resolved.

User Maintenance

This section is broken out into the following tasks:

- **Adding a New User**
  To add a new user, please refer to the following section entitled Adding a New NatQuery User

- **Modifying an Existing User**
  To modifying information pertaining to an existing NatQuery user, please refer to the following section entitled Modifying an Existing NatQuery User.

- **Deleting an Existing User**
  To delete an existing NatQuery user, please refer to the following section entitled Deleting an Existing NatQuery User.

After making any modification to the users of NatQuery, and in situations where NatQuery is being used by End-Users, the Administrator will need to "Roll-Out" the Environment Configuration as outlined in the section entitled Rolling Out an Environment Configuration.

Adding a New NatQuery User

To add a new user to NatQuery, perform the following steps:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Instructions on how to start NatQuery can be found in the section entitled Starting NatQuery.

2. **Invoke the Administer Users function**
   On the NatQuery desktop, and with no query open, click on the **Administer** drop-down menu, then click on **Environment Configuration / Server Connection Configuration / User Information**. This will invoke the **Administer Users** function.

   Specific help on how to utilize the **Administer Users** function is available if you click the **Help** button while on this form.

3. **Add User(s)**
   Clicking the **New** button on the Administer window will bring up the **Administer User Information** window. If a default user has been set up (a recommended action during the initial install of NatQuery), then the only information necessary will be the user's name and user ID. Otherwise, all information will have to be entered manually.

   Once all information has been entered, clicking the **OK** button will store the new user information and return the user to the **Administer Users** window.

   Specific help on how to utilize the **Administer User Information** function is available if you click the **Help** button while on this form.

   Clicking **Cancel** on this window will cause all changes to be lost and return you to the **Administer Users** window.

4. **Verify Environment Configuration**
   Once all users that need to be added are acted upon, clicking the **OK** button on the **Administer Users** window will close this function.

   As this window closes, NatQuery will prompt with a message indicating that the changes made require the Verify Environment Configuration function to be run, and it will ask if this function should be performed. When presented with this prompt, it is suggested that the Administrator click **Yes** so the Verify Environment Configuration function will execute.

5. **Roll-Out the new Environment Configuration**
   In situations where the Environment Configuration should no be given to users, the Administrator should consider performing the steps necessary to "Roll-Out" the changed Environment Configuration. Instructions for Rolling-Out an Environment Configuration

can be found in the section entitled Rolling Out an Environment Configuration.

Modifying an Existing NatQuery User

To modify information about an existing user of NatQuery, perform the following steps:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Instructions on how to start NatQuery can be found in the section entitled Starting NatQuery.

2. **Invoke the Administer Users function**
   On the NatQuery desktop, and with no queries open, click on the **Administer** drop-down menu, then click on **Environment Configuration / Server Connection Configuration / User Information**. This will invoke the **Administer Users** function.

   Specific help on how to utilize the **Administer Users** function is available if you click the **Help** button while on this form.

3. **Select User(s) to Modify**
   From the **Currently Defined Users** list box that is available on the **Administer Users** function, find and then select the user whose information should be modified by clicking on the user in this list. To modify the current information for this user, click on the **Edit** button.

   This action will invoke the **Administer User Information** window, which will display information that pertains to the selected user. Perform whatever modifications are required.

   Once the desired information has been modified, clicking the **OK** button will store the modified user information and return the user to the **Administer Users** window.

4. **Verify Environment Configuration**
   Once all users that need to be modified are acted upon through the **Administer Users** function, clicking the **OK** button will close this function.

   As this window closes, NatQuery will prompt with a message indicating that the changes made require the Verify Environment Configuration function to be run, and it will ask if this function should be performed. When presented with this prompt, it is suggested that the Administrator click **Yes** so the Verify Environment Configuration function will execute.

5. **Roll-Out the new Environment Configuration**
   In situations where the Environment Configuration should no be given to users, the Administrator should consider performing the steps necessary to "Roll-Out" the changed Environment Configuration. Instructions for Rolling-Out and Environment Configuration can be found in the section entitled Rolling Out an Environment Configuration.

Deleting an Existing NatQuery User

To delete information about an existing user of NatQuery, perform the following steps:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Instructions on how to start NatQuery can be found in the section entitled Starting NatQuery.

2. **Invoke the Administer Users function**
   On the NatQuery desktop, and with no queries open, click on the **Administer** drop-down menu, then click on **Environment Configuration / Server Connection Configuration / User Information**. This will invoke the **Administer Users** function.

   Specific help on how to utilize the **Administer Users** function is available if you click the **Help** button while on this form.

3. **Select and Delete User(s)**
   From the **Currently Defined Users** list box that is available on the **Administer Users** function, find and then select the user that should be deleted by clicking on the user in this list.

   Clicking the **Delete** button will delete the selected user.

4. **Verify Environment Configuration**
   Once all users that need to be deleted are acted upon through the **Administer Users** function, clicking the **OK** button can close this function.

   As this window closes, NatQuery will prompt with a message indicating that the changes made require the Verify Environment Configuration function to be run, and it will ask if this function should be performed. When presented with this prompt, it is suggested that the Administrator click **Yes** so the Verify Environment Configuration function will execute.

5. **Roll-Out the new Environment Configuration**
   In situations where the Environment Configuration should no be given to users, the Administrator should consider performing the steps necessary to "Roll-Out" the changed Environment Configuration. Instructions for Rolling-Out an Environment Configuration can be found in the section entitled Rolling Out an Environment Configuration.

DDM Maintenance

There are many aspects to DDM Maintenance, with each of these task outlined below.

After performing any DDM Maintenance, and in situations where the Administrator is making changes that are intended to be subsequently used by one or more end-users, the Administrator will likely have to "Roll-Out" the changed Environment Configuration, of which the DDM changes are a part, prior to these changes being seen by an End-User. For instructions for Rolling-Out and Environment Configuration can be found in the section entitled Rolling Out an Environment Configuration.

This various aspects of DDM Maintenance are broken out into the following sections / tasks:

- **Adding a New DDM**
  To add a new DDM, please refer to the following section entitled Adding a New DDM.

- **Modifying an Existing DDM**
  To modifying information pertaining to an existing NatQuery DDM, please refer to the following section entitled Modifying An Existing DDM.

- **Deleting an Existing DDM**
  To delete an existing NatQuery DDM, please refer to the following section entitled Deleting An Existing DDM.

- **Occurrence Information**
  Occurrence Information is required information for any DDM that contains any recurring fields such as Multi-Valued Fields (MUs) or Periodic-Groups (PEs). To handle **Occurrence Information** for a DDM, please refer to the following section entitled Occurrence Information.

- **Descriptor Statistics Information**
  Descriptor Statistics Information is required for any DDM that contains any recurring fields such as Multi-Valued Fields (MU) or Periodic-Groups (PE). To handle **Descriptor Statistics Information**, please refer to the following section entitled Descriptor Statistic Information.

- **File Relationship Information**
  File Relationship Information is required for any situation where a single extraction process will be expected to extract data from more than one source file.

- **Field Sign Byte Information**
  Field Sign Byte Information is required for any situation where a numeric source field may be a negative value, such that a sign byte should automatically be supplied to the field whenever the field is referenced for extraction. Information on assigning Sign Bytes to Numeric fields can be found in the section entitled Field Sign Byte Information.

- **Field Edit Masks**
  Field Edit Masks allows for an Administrator to assign Edit Masks that will be used when fields are output by NatQuery or NatCDC. Information on handling Field Edit Masks can be found in the section entitled Field Edit Mask Information

- **Redefine DDM Fields**
  Based on the use of fields on the server, single fields may actually represent multiple fields as a result of an application redefining DDM fields into components. To address this, the Administrator may redefine fields at the DDM level, allowing users to see the redefined components as opposed to the single field. Please refer to the section entitled Redefine DDM Fields.

- **File Variables**
  In specific situations, an Administrator may need to create variables that are directly associated with a DDM. Please refer to the section entitled File Variables.

- **DDM Editor**
  If a DDM needs to be manually manipulated, this can be accomplished using the NatQuery DDM Editor. Please refer to the section entitled DDM Editor.

After making any modification to an Environment Configuration, and in situations where NatQuery is being used by End-Users, the Administrator will need to "Roll-Out" the Environment Configuration as outlined in the section entitled Rolling Out an Environment Configuration.

Adding a New DDM

Adding a new DDM to NatQuery is the first step of making a new DDM available for NatQuery processing.

There are basically two approaches to adding a new DDM into a NatQuery Environment Configuration.

The first approach to adding a new DDM is to import a new DDM The DDM file to be imported can either be automatically created by a NatQuery-spawned process and then downloaded and automatically imported through NatQuery, or the DDM can be provided into the NatQuery environment through some other fashion and then imported. The DDM Import function of NatQuery can import a DDM contained in a SYSTRANS format, SYSOBJH format, or in one of the several “standard” DDM text layouts that SAG supports / provides. Whenever NatQuery needs to become aware of a DDM that already exists somewhere else, the import approach to adding a new DDM into NatQuery is the superior approach, and in most cases this process is almost completely automatic.

The second approach is to use a function of NatQuery that provides a basic DDM editor. Using this DDM editor, a new DDM can be created, or an existing DDM can be easily modified. This approach is an atypical approach however, and is usually used when no DDM currently exists for a file (such as what may be the case for a sequential file created by some external process) that NatQuery needs to read. If using the DDM editor is deemed by the Administrator to be the best approach to adding a new DDM, then the Administrator should forgo this specific section in favor of the section entitled DDM Editor.

Assuming that the Administrator would like to import a new DDM into NatQuery, then the file to be imported can be provided in several ways:

- **Automatically Capture DDMs through NatQuery**
  As this is the preferred method, this approach will be described below.

- **Capture DDMs through Manual Execution of SYSTRANS / SYSOBJH**
  Manual execution of SYSTRANS or SYSOBJH can result in an output file that contains DDM information. This output can then be brought down to the workstation in several manual ways (FTP, IND$FILE, Copy operations). Once on the workstation, this file can then be imported into NatQuery using the Import DDM function. The DDM Import function is found by clicking on Administer / Environment Configuration / DDMs-FDTs / Import DDM.

- **Capture DDMs through Entire Connection**
  If the Software AG product Entire Connection is present in your environment, then DDMs can be downloaded to the workstation using a function of that software. Once downloaded via Entire Connection these DDMs can then be imported into NatQuery using the Import DDM function.

- **Manually create DDMs using the Build / Edit DDM Function**
  Using a function of NatQuery that is designed to allow for DDMs to be built or edited, it is possible to manually enter the structure of a DDM directly into NatQuery using a listing of a DDM as a basis. This approach is only recommended in unusual situations.

- **Capture DDMs through listings with Cut & Paste**
  As a method of last resort, a DDM can be built using a text editor such as Notepad, along with a terminal emulation session by listing a DDM file on the server environment, and then using Cut & Paste between the emulation session and the text editor, this technique can result in a DDM file that can be imported into NatQuery.

As it is highly desirable to configure NatQuery so that it will automatically handle DDM download requests, the following instructions will describe the steps to be taken to accomplish this, with these steps utilizing the **Download DDM** function of NatQuery.

If the **Download DDM** function is not operational for whatever reason, then the reader may refer to the Capture DDMs step of the section entitled Step 6 - Environment Configuration which describes how to make this function operational. Alternatively, the reader may use one of the more manual approaches outlined briefly above to introduce a new DDM into NatQuery.

IMPORTANT NOTE #1:
Prior to moving forward with the capture of DDMs, the Administrator should insure that the DDMs that are to be captured have been generated with the option that will detail the components of Super-Descriptors and Sub-Descriptors within the DDMs. An example of this would be (note bolded text):

```
TYL  DB  NAME                              F LENG  S D REMARKS
---  --  --------------------------------  - ----  - - -------------------------


  1  H1  LEAVE-LEFT                        B    4  N S
*        -------- SOURCE FIELD(S) -------
*        LEAVE-DUE(1-2)
*        LEAVE-TAKEN(1-2)
  1  S1  DEPARTMENT                        A    4    S
*        -------- SOURCE FIELD(S) -------
*        DEPT(1-4)
* 1  S2  DEPT-PERSON                       A   26    S
*        -------- SOURCE FIELD(S) -------
*        DEPT(1-6)
*        NAME(1-20)
```

Having this information in DDMs is a requirement for NatQuery in any situation where it is expected to use “selective extraction” as opposed to extracting entire files.

IMPORTANT NOTE #2:
If possible, it is desirable that the DDMs that will be given to NatQuery be generated so that occurrence information exists in the “Remarks” column of the DDMs. An example of this would be (note bolded text):

```
TYL  DB  NAME                             F LENG  S D REMARKS
---  --  -------------------------------- - ----  - - ------------------------

M 2  AI  ADDRESS-LINE                     A   20  N   Max. occurrences 3
         HD=ADDRESS
```

There is no requirement that such occurrence information exist in a given DDM, however if this information is present and is in the format as shown above, NatQuery can automatically import this information.

To add a new DDM to NatQuery using the **Download DDM** function, perform the following steps:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Information on starting NatQuery can be found in the section entitled Starting NatQuery.

2. **Insure Environment Path is Correct**
   A given **Environment Configuration** is a collection of files that reside in the same physical directory, with this directory commonly referred to as the **Environment Path**. This **Environment Path** will contain the information built in previous steps, and will also be the container for information related to NatQuery generation.

   During the initial configuration of NatQuery, it is suggested that the first Environment Configuration for a given site should be the **FILES** subdirectory of the NatQuery install directory.

   Subsequent to the initial install and configuration, an Administrator may wish to have NatQuery serve as a single point of administration point for several different Environment Configurations.

   Either way, the current setting of the NatQuery Environment Configuration Path is shown on the NatQuery Toolbar beside the label “Environment Path”.

   If the Administrator needs to change the current setting of Environment Path to some other directory, then this is accomplished by changing the value of the field labeled Environment Path which is found on the Environment Paths tab of the NatQuery Configuration function. This function is accessed from an empty NatQuery desktop by clicking the Administer drop-down menu, and then clicking NatQuery Configuration.

3. **Utilize Download DDM function**
   If NatQuery has been configured with the **Server Communication Mode** set to be **FTP** or **PC Network**, then NatQuery provides a function that will generate the JCL or Script that will handle the extraction of the DDMs required through the execution of the Natural SYSTRANS or SYSOBJH utilities in batch. It will additionally attempt to move the generated JCL or Script onto the server.

To use NatQuery's ability to automatically capture DDMs through SYSTRANS or SYSOBJH, perform the following:

3.1. **Invoke Download DDM Function**

Click on the **Administer** drop-down menu. On the resulting sub-menu, select **Environment Configuration / DDM(s)/FDT(s) / Download DDM(s)**. This will invoke the **Download DDM** window.

Using the **Download DDM** window, enter the name(s) of the DDM(s) of interest. Since DDMs can be easily introduced later, *it is suggested* that the Administrator start by providing a single DDM name only. This is because it can be expected that it may take several iterations to get the **Production Special Process** JCL / Script template fully operational, and for initial iterations it is easier to handle a single DDM. Once the Production Special Process is refined / tweaked so that it will run dependably – other DDMs can be easily downloaded later.

Detailed information on how to use the **Download DDM** window is available by pressing the **Help** button while in this function.

When all desired DDM names have been entered into the **Download DDM** window, click the **OK** button.

If NatQuery is configured to use **FTP** and NatQuery has not yet performed an **FTP** operation within the NatQuery session, then NatQuery will respond by providing a **FTP Password** window that prompts for the user's **FTP** password similar to Figure 5 (**FTP Password Question**). If this is the case, then the user should provide the appropriate **FTP** password and then click the **OK** button on the **FTP Password** message box.

If NatQuery has previously performed a successful **FTP** operation within the same NatQuery session, or the user just provided the appropriate **FTP** password to the preceding prompt, then the user will be prompted for confirmation of the **FTP** operation, and the user should click the **OK** button.

If NatQuery is configured to use **PC Network**, the user will be prompted for confirmation of the **PC Network** Copy operation, and the user should click the **OK** button.

Through either an **FTP** operation or a **PC Network** Copy operation, the JCL or Script to handle the execution of SYSTRANS or SYSOBJH that will extract the specified DDMs from the Natural server platform should now have been placed on the server.

If NatQuery encounters errors with either **FTP** or **PC Network** Copy operation, then the Troubleshooting section of this manual should be referred to, and once corrective action has been taken this entire step should be re-run.

If NatQuery is integrating against a mainframe Natural server, then it is typical that the

NatQuery will be configured to use **Direct FTP**. In this case, the JCL should have been placed into JES (on MVS platform) or POWER (on a VSE platform).

If NatQuery is integrating against a non-mainframe Natural server and **Remote Execution** is being used to remotely execute queries, and errors develop in this **Remote Execution** processing, then the Troubleshooting section of this manual should be referred to. Once corrective action has been taken this entire step should then be re-run.

Depending upon how integration to the Natural Server has been configured, the JCL or Script request may require manual intervention to have this JCL or Script be physically executed on the server. If such is the case, then this manual intervention should be performed now. If any question arises as to how to execute the JCL or Script then the reader should refer to the section entitled JCL / Script Template.

If NatQuery encounters errors with the JCL or Script execution, then in most cases the problem will typically be resolved through changes to the **Production Special Process** JCL / Script template. In this case, the changes needed should be made to the **Production Special Process** JCL / template, with this entire step being re-run. Specific information on handling the **Production Special Process** JCL / Script template can be found in the section entitled JCL / Script Template, and information on correcting typical execution errors can be found in the Troubleshooting section of this manual.

Once execution of the request to download DDMs has been executed, then the **Check Server** step can be performed.

3.2. Check Server

When the **Production Special Process** JCL or Script has been executed successfully, the Administrator should be able to download the resulting SYSTRANS or SYSOBJH output file into NatQuery by utilizing the **Check Server** function. To use the **Check Server** function, click on the **Check Server** icon found on the NatQuery Toolbar: This action will invoke the **Check Server** window.

When the **Check Server** window is initially presented, it will provide information concerning the “Request Slots” that have been assigned to the User ID being used, as well as the current status of these. If defaults were applied to the NatQuery definition of the User, the number of Request Slots will be 3 (three) unless administratively over-ridden.

If the preceding instructions have been followed, then (at least) one of these requests should show itself to be a “**SYSTRANS - Download DDM Request**” or “**SYSOBJH – Download DDM Request**” (most likely as Request #1), and this request should show a status of “**PENDING**”.

The information concerning the displayed request(s) will be initially pulled from the user’s **Local Log File,** which is the file resident within the workstation environment that is updated directly by NatQuery as it submits requests, such that the **Local Log File** tracks what a user has sent for execution.

Continue to the next step.

3.3. **Check Server For Update – Check Server**
As a result of the **Production Special Process** JCL or Script executing, the Natural program NQYP0003 (mainframes) or NQYPNT03 (open systems) should have been executed, and this execution should update the user's **Remote Log File**. The **Remote Log File** is a file that is resident within the Natural server environment and is updated by JCL or Script processing, such that the **Remote Log File** tracks the execution status of a user's request on the server platform.

In the lower left of the **Check Server** window, there is a button labeled **Check Server for Update** button: The user should now click on this button.

This action will cause NatQuery to attempt to retrieve the user's **Remote Log File** from the server onto the workstation platform using either **FTP** or a **PC Network** Copy operation.

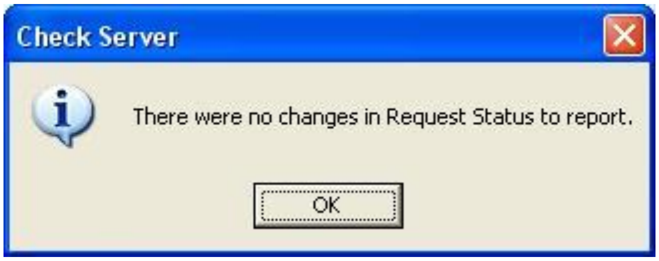
If a problem develops with the **FTP** or **PC Network** copy operation of the user's **Remote Log File** from the Natural server platform to the workstation environment, then the user should refer to the Troubleshooting section of this manual. Once corrective action is taken, this entire step should be re-done.

If the **FTP** or **PC Network** copy operation is successful, then the information displayed in the **Log Status** frame of the **Check Server** window for the specific request slot should change from "**PENDING**" to either "**DONE**" or "**FAILED**".

If the request shows as "**DONE**", then the user may continue to the next step.

If the request shows as "**FAILED**", then there was a problem with the execution of the request JCL or Script, and this problem will most likely be corrected through changes to the **Production Special Process** JCL / Script template. In this case, the user should refer to the JCL / Script Handling section of this manual for further information on handling this template. Additional information can also be found on corrective action by reviewing the Troubleshooting section of this manual. Either way, once corrective action has been taken this entire step should then be re-done.

If NatQuery reacts to the **Check Server For Update** operation by displaying a message box similar to the following,



Check Server

There were no changes in Request Status to report.

Figure 7 – No Changes To Report

then this indicates that the **Remote Log File** has not been properly updated by the execution of the JCL or Script. If this is the case, then it may be that the request has not physically executed yet, and this fact should be verified. If the request did execute, then it is likely that the processing failed in a severe fashion such that the user's **Remote Log File** was not updated either with a "**DONE**" or "**FAILED**" message. In this case the Troubleshooting section should be referred to and once corrective action is taken then this step should be fully re-done. If it is not yet executed, steps should be taken to have this JCL or Script get executed. If any question arises as to how to execute the JCL or Script, then the reader should refer to the section entitled JCL / Script Template.

3.4. **Retrieve Request Output**

If after clicking the **Check Server for Update** button NatQuery responds by updating the status of the DDM request from "**PENDING**" to "**DONE**", then the request has been executed and the DDM output can now be downloaded into NatQuery.

This is accomplished by first clicking anywhere on the text of the request that now shows as "**DONE**". This action will highlight all text associated with that request and it will additionally enable the buttons labeled **Retrieve Request Output** as well as **Clear Selected Request**.

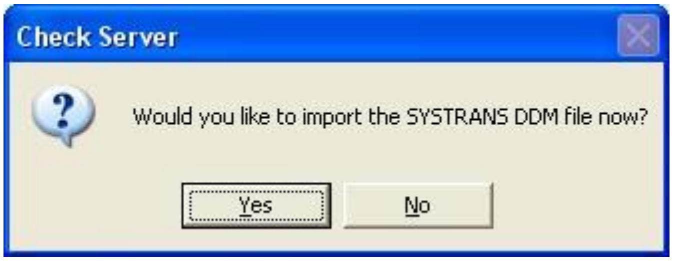
By clicking on the **Retrieve Request Output** button, NatQuery will attempt to retrieve the DDM request's output file from the Natural server platform onto the NatQuery workstation platform using either **FTP** or a **PC Network** Copy operation.

If a problem develops with the **FTP** or **PC Network** copy operation of the output file from the Natural server platform to the workstation environment, then the user should refer to the Troubleshooting section of this manual. Once corrective action is taken, this entire step should be fully re-done.

3.5. **Automatically Import SYSTRANS / SYSOBJH File**

As a result of the above step, NatQuery will have invoked an FTP or PC Network copy operation that should retrieve the SYSTRANS or SYSOBJH output from the Natural Server and place this file onto the workstation.

When the FTP or PC Network copy operation is completed, the user will be presented with a message box similar to the following.



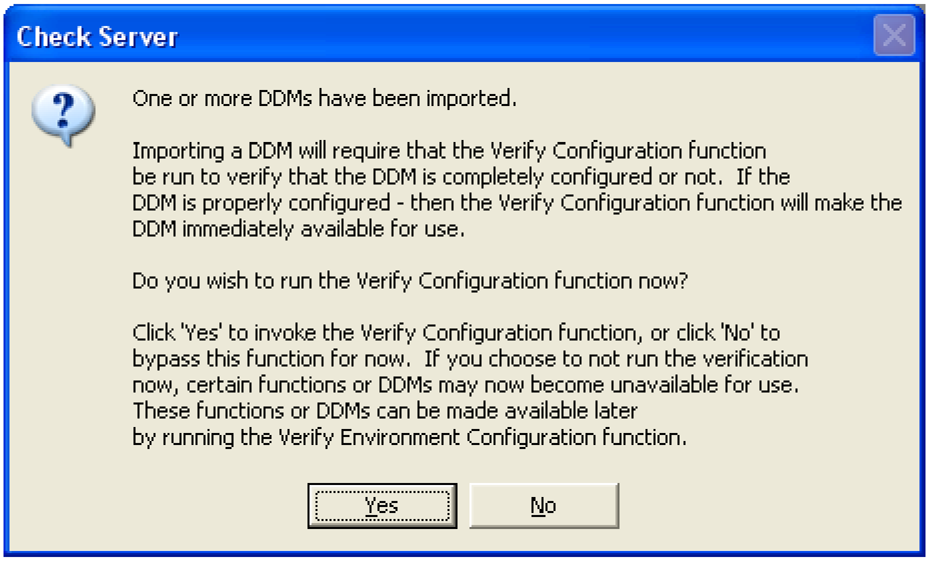
Check Server

Would you like to import the SYSTRANS DDM file now?

Figure 8 – **Import Question**

The user should click **Yes** to this prompt. NatQuery will respond to this by internally performing the work necessary to import the DDM(s), and after the import is completed the user will be returned to the **Check Server** window.

The user can then proceed by clicking the **OK** button or the **Cancel** button to close the Check Server window. As this window closes, a message box will be shown that is similar to the following:



Check Server

One or more DDMs have been imported.

Importing a DDM will require that the Verify Configuration function be run to verify that the DDM is completely configured or not. If the DDM is properly configured - then the Verify Configuration function will make the DDM immediately available for use.

Do you wish to run the Verify Configuration function now?

Click 'Yes' to invoke the Verify Configuration function, or click 'No' to bypass this function for now. If you choose to not run the verification now, certain functions or DDMs may now become unavailable for use. These functions or DDMs can be made available later by running the Verify Environment Configuration function.

Figure 9 – **DDM Import / Verify Question**

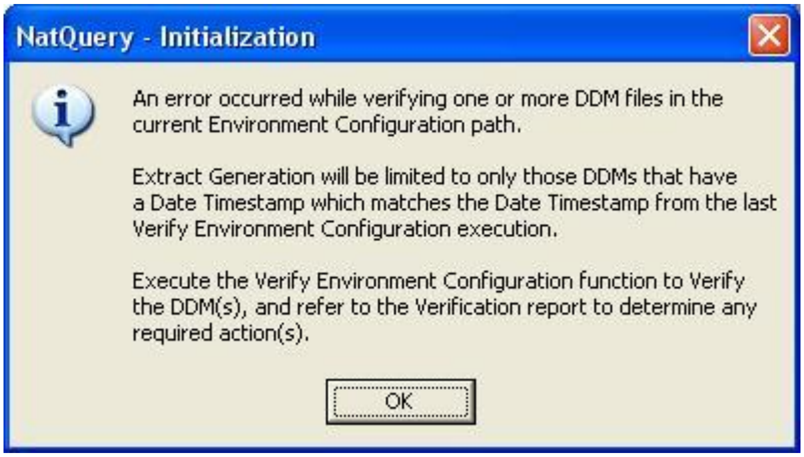
In response to this message, the user may click either the **Yes** button or the **No** button, however *it is suggested* that the user click **Yes**. Clicking **No** will simple close the **Check Server** window, returning the Administrator immediately to the NatQuery desktop. Clicking **Yes** will cause NatQuery to validate the DDM(s) that were just imported.

When handling a new DDM, the Verify process will detail the further actions required to allow the DDM to become enabled for NatQuery processing.

Further information on the NatQuery **Verify Environment Configuration** function can be found in the section entitled Verifying an Environment Configuration.

Closing the **Verify Configuration** window will return the Administrator to the NatQuery desktop.

In the process of being returned to the NatQuery desktop, the use may see a message similar to the following:



An error occurred while verifying one or more DDM files in the current Environment Configuration path.

Extract Generation will be limited to only those DDMs that have a Date Timestamp which matches the Date Timestamp from the last Verify Environment Configuration execution.

Execute the Verify Environment Configuration function to Verify the DDM(s), and refer to the Verification report to determine any required action(s).

Figure 10 – **Initialization Error**

This error should be considered normal, as at this point the Verify process will likely find that any new DDM(s) just imported does not have all of the related information needed. In response to this message, the user can click the **OK** button.

When the Administrator is back on the NatQuery desktop, the Administrator will now typically proceed to act upon the information seen in the Verify Configuration report as to what additional work need to be done to allow a given DDM to become Verified (I.E. address either Occurrence Information and / or Descriptor Statistic Information for the given DDM).

Modifying An Existing DDM

There are basic two approaches to modifying a DDM into a NatQuery Environment Configuration.

The first approach is to modify an existing NatQuery Configuration DDM is by importing a newer / changed version of the DDM. The DDM file to be imported can either be automatically created by a NatQuery-spawned process and then downloaded and automatically imported into NatQuery, or the newer version of the DDM can be provided into the NatQuery environment through some other fashion and then imported. The DDM Import function of NatQuery can import a DDM contained in a SYSTRANS format, SYSOBJH format, or in one of the several “standard” DDM text layouts that SAG supports / provides. Whenever NatQuery needs to have an existing DDM be updated with a newer / changed version that already exists somewhere else, the import approach to modifying an existing DDM is the superior approach, and in most cases this process is almost completely automatic.

The second approach is to use a function of NatQuery that provides a basic DDM editor. Using this DDM editor, a new DDM can be created, or an existing DDM can be modified. This approach is an atypical approach however, and is typically only used when no DDM currently exists for a sequential file that NatQuery may then read / process. If using the DDM editor is deemed by the Administrator to be the best approach to modifying a new DDM, then the Administrator should forgo this specific section in favor of the section entitled DDM Editor.

To modify an existing DDM that exists somewhere else, then the instructions to accomplish this are essentially the same as that found in the section entitled Adding a New DDM, and the Administrator should therefore refer to that section.

Deleting An Existing DDM

To delete a DDM that exists in the NatQuery environment, perform the following steps:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Information on starting NatQuery can be found in the section entitled Starting NatQuery.

2. **Utilize Delete DDM function**
   On an empty NatQuery desktop, click on the **Administer** drop-down menu, then click on **Environment Configuration / DDMs-FDTs / Delete DDM**. This will invoke the **Delete DDM** function. Specific help on using the **Delete DDM** function can be found by clicking the **Help** button while in this function.

   Using the selection box, locate the DDM to be deleted and then select this DDM by clicking on it – this action should highlight the DDM's name. If the name of the DDM requiring deletion cannot be found in the selection box, then the DDM does not exist in the current Environment Configuration path. This may mean that the DDM was never imported, or that the current setting of the Environment Path needs to be changed.

   If the DDM is found, click the **Delete** button. NatQuery will respond by requesting confirmation of the Delete DDM request, and it is expected that the user will click the **Yes** button in response.

   If the DDM being deleted happens to have been either partially or even fully configured, then NatQuery will make an additional prompt to request permission to delete the administrative information in addition to deleting the DDM file. If the intent of deleting the DDM is to remove it temporarily from an Environment Configuration but to later re-instate the DDM (or another "version" of this DDM), then it may be desirable to not delete this information. In most cases, it is typically expected that any related Configuration Information should also be deleted at the time the DDM is deleted, so the user would typically click **Yes** to the prompt asking about deleting related Administrative Information. If the Administrator is unsure of what to do, clicking the **Cancel** button will cancel the deletion of the DDM.

   Subsequent to internally deleting the DDM as well as optionally deleting any configuration information related to that DDM, the Administrator will be returned to the **Delete DDM** window.

   When all DDMs requiring deletion have been handled, the Administrator will click the **OK** button to close the DDM window.

   As the Delete DDM window closes, and depending on the attributes of the DDMs that were deleted, NatQuery may prompt the Administrator with a message that asks if a Verify Environment Configuration should be performed. It is strongly advised to click **Yes** to the prompt, review the resulting report for correctness, and then close the Verify

Configuration window by clicking the **OK** button.

Further information on the NatQuery Verify Environment Configuration function can be found in the section entitled Verifying an Environment Configuration.

Occurrence Information

In order to allow any given DDM to be verified for subsequent handling by NatQuery, **Occurrence Information** must be provided for any recurring fields such as Periodic-Group fields (PEs) or Multi-Value fields (MUs) that may exist in a DDM. The **Occurrence Information** that will be provided to NatQuery for both PEs and MUs will be the maximum number of occurrences that should be referenced for any given recurring field (this is usually application controlled), as well as the default number of occurrences to be referenced by default (which will generally be the same number as the maximum number).

When a user selects any given recurring field for handling, NatQuery will utilize **Occurrence Information** to automatically provide the default number of occurrences, thus shielding the End-User from having to definitively know what occurrences should be selected for retrieval.

Handling **Occurrence Information** for a given DDM is only required in situations where a given DDM contains MU or PE fields. If the Administrator is sure that a given DDM contain no MU or PE fields, then **Occurrence Information** handling can be bypassed for that DDM.

For each DDM imported that does contain MU or PE fields, **Occurrence Information** must be provided for these fields prior to NatQuery allowing the given DDM to be Verified for use.

For each file that requires **Occurrence Information** as a result of containing one or more MU and / or PE fields, one of three situations will likely exist:

- **Occurrence Information** is unknown and needs to be definitively determined, the Administrator will need to insure that accurate information is provided. If this is the case, then the reader should proceed with Step #1 immediately following.

- **Occurrence Information** is documented in the DDM through a Predict Generation option that placed a “Max. occurrences xx” comment in the Remarks column of the DDM. If this is the case, then the Administrator can move to Step #2 following below.

- **Occurrence Information** for all recurring fields for a given DDM is definitively known by the Administrator and will need to be manually entered. If this is the case, then the user can proceed with Step #2 following below.

1. **Determine Occurrence Information Programmatically**
   Through a function of NatQuery, a Natural program can be generated and submitted to the server that will programmatically determine the maximum number of occurrences used by any recurring field in a file. This approach to gathering the required **Occurrence Information** can either be accomplished based on a sampling of records from the source file in question, or it can be based upon all records in the DDM file.

   This functionality is encapsulated into a **Data Discovery Function** of NatQuery, specifically the **Repeating Field Analysis** function. On an empty NatQuery desktop, this function is accessed by first clicking on the **Administer** drop-down menu, and then clicking on

Environment Configuration / Data Discovery – Analysis / Repeating Field Analysis. These actions will invoke the **Analyze Repeating Fields** window, and specific help for using this function being available by clicking the **Help** button while in this function.

At the top left of the **Analyze Repeating Fields** window is a combo box that allows the file of interest to be selected. If the name of the DDM is not listed in this combo box, then the DDM in question contains no recurring fields (either MUs, PEs or MUs in PEs). If this is the case, then the reader may skip handling **Occurrence Information** for the given DDM. If the DDM name does show in the combo box, then this DDM should be selected by clicking on its name (I.E. highlighting it).

2. Subsequent to selecting the name of the DDM, NatQuery will provide defaults that will route the final output to Excel, will run the analysis against all records, and will run the analysis against all recurring fields. Generally, these defaults are acceptable, but can be changed as necessary. When the options have been reviewed and any needed changes to the default options have been made, the user will click the **OK** button on the **Analyze Repeating Fields** window. This action will cause NatQuery to generate the Natural program required to capture the desired information from the specified file, with this program being encapsulated into the **Production Request Process** JCL / Script template. Depending upon the method of integration between the NatQuery workstation and the Natural server platform, this process may be executed automatically against that Natural server platform or not.

Once the **Analyze Repeating Fields** request for the first file has been moved to the server, the Administrator should then close the **Analyze Recurring Fields** function by clicking the **Cancel** button while in this function.

For the initial execution of the **Analyze Repeating Fields** function, the Administrator is advised to manually check the status of the submitted request, and take whatever action is necessary to insure the request gets executed if this is necessary. Depending upon the method of integration between NatQuery and the Natural server platform, this may require manual intervention to insure that the request gets executed, and / or it may also require correction action to the Production Request Process template to properly execute.

If problems develop in moving the request to the Natural server, then the user should refer to the Troubleshooting section. If questions arise on the functioning of the **Production Request Process** template or how this can be executed, then the reader should refer to the JCL / Script Template section of this manual.

If the request did not execute successfully, then corrective action should be taken (typically this will require changes to the **Production Request Process** template), and this step should then be re-done.

If the request did execute successfully, then this fact should be seen by using the **Check Server** function – this function is invoked by clicking on the **Check Server** icon found on the NatQuery toolbar. By invoking this function, and then clicking the **Check Server For Update** button, the Administrator should see the status of the **Repeating Field Analysis** request change from “**PENDING**” to “**DONE**” or “**FAILED**”.

If the status shows as “**FAILED**”, then the likely cause will be a problem with the execution of the **Production Request Process** JCL or Script. In this case, corrective action will most likely be needed with the **Production Request Process** JCL / Script template, after which this step will need to be re-run. Information on the handling of this template can be found by referring to the JCL / Script Template section of this manual; information on common errors may be found in the Troubleshooting section of this manual.

If the status does not change at all, but rather a message similar to Figure 7 (**No Changes To Report**), then either the request has not yet been executed (in which case steps should be taken to insure its execution), or the process failed to the degree where no update whatsoever was able to be made to the user’s Remote Log file. In this latter case, corrective action will most likely be needed with the **Production Request Process** JCL / Script template, after which this step will need to be re-run. Information on the handling of this template can be found by referring to the JCL / Script Template section of this manual; information on common errors may be found in the Troubleshooting section of this manual.

If the status changes to “**DONE**”, then the request executed properly and the resulting data can be downloaded. This is accomplished by clicking anywhere on the text of the message associated the request slot that shows as “**DONE**”, and then clicking the **Retrieve Request Output** button. This action will cause NatQuery to initiate the process of moving the output of the **Repeating Field Analysis** from the Natural server platform onto the workstation.

NatQuery will begin this process by presenting a window named **Save Extract Data File**, and a default file name of “**ANALYSIS.TXT**”; *it is suggested* that this default name be accepted by clicking the **Save** button.

NatQuery should then perform the work necessary to download the file, and when the process is complete a message will be provided that indicates completion of this task. The Administrator should click **OK** to close this message box.

If, when the **Repeating Field Analysis** prompt was initiated, the target of the analysis was indicated to be Excel, then NatQuery should prompt the user if Excel should now be invoked to display the resulting output. In such case, the user may click **Yes** to invoke Excel (then review the data in Excel, possibly save this and then close Excel), or may click **No** and bypass this invocation.

Either way, the user will end up back on the **Check Server** window, where the downloaded request will now be cleared and the corresponding request slot will now show as “**OPEN**”. The user may now close the **Check Server** window by clicking the **OK** button or by clicking the **Cancel** button.

In response to this, NatQuery may provide a message similar to that seen in Figure 10 (**Initialization Error**). When adding brand new DDMs into NatQuery, this error should be considered normal as at this point the Verify process will most likely find that a DDM just imported does not have all of the related information needed to allow this DDM to become usable by NatQuery for extraction processing. In response to this message, the user can click the **OK** button.

3. **Invoke Administer Occurrence Information Function**
The **Administer Occurrence Information** function is invoked from an empty NatQuery desktop by clicking on **Administer / Environment Configuration / Occurrence Information**. These actions will invoke the **Administer Occurrence Information** window, with help on using this function being made available by clicking the **Help** button while in this function..

With the **Administer Occurrence Information** presented, the DDM requiring **Occurrence Information** should be selected by using the combo box in the upper left corner. If the name of the DDM requiring Occurrence Information cannot be found in the combo box, then the DDM does not exist in the current Environment Configuration path. This may mean that the DDM was never imported, or that the current setting of the Environment Path needs to be changed.

If the Administrator wishes to import the results obtained by running the **Repeating Field Analysis Data Discovery** function from the preceding step, then the act of selecting the DDM name will enable a button in the upper right-hand corner named **Import From Repeating Field Analysis**. By clicking the **Import From Repeating Field Analysis** button, NatQuery will import the results of the **Repeating Field Analysis** and will apply these values to the grid.

If the Administrator is performing this step with the intent of manually entering the required **Occurrence Information**, then the Administrator may now proceed to do this by using the graphical controls on the **Administer Occurrence Information** window.

If the Administrator is performing this step as a result of knowing that a given DDM contains occurrence information in the Remarks section of the DDM, then this DDM information should be reflected in the Default column of the central grid control with the text “DDM Default”. Values of DDM Default will need to be individually reviewed and modified from the “Default” value to a definitive selected value(s) prior to the Occurrence Information being accepted by NatQuery.

The general processing of the **Administer Occurrence Information** window is such that when a DDM is selected, the first recurring field found in the DDM will be automatically selected for handling. With a specific recurring field selected, the graphical controls in the lower portion of the **Administer Occurrence Information** window will become enabled to allow changes to be made for that recurring field. Once necessary changes are made to that recurring field, the user will click the **Apply** button. Clicking the **Apply** button will move the contents of the graphical controls into the grid, and NatQuery will then automatically select the next recurring field in the grid for handling.

When valid values / changes have been applied to all recurring fields for the given file, and the entries all appear correct, the **Apply Changes** button will become enabled.

Clicking the **Apply Changes** button will cause NatQuery to edit the contents of the window / grid. If all is satisfactory, NatQuery will internally save this information and it will then clear the **Administer Occurrence Information** window of all information related to the just-handled DDM in preparation for the possibility of handling another DDM.

If other DDMs exist that need to have Occurrence Information handled, then these can be handled now by repeating the procedure described above.

When all DDMs that require Occurrence Information have been handled, the Administrator will click the **OK** button to close the **Administer Occurrence Information** window.

As the **Administer Occurrence Information** window closes, a message will be displayed similar to Figure 3 (**Verification Question**).

It is suggested that the Administer click **Yes** to this prompt, and review the results of the Verify Configuration report as this report will guide the Administrator if any further action is necessary. Further information on the NatQuery **Verify Environment Configuration** function can be found in the section entitled Verifying an Environment Configuration.

Closing the **Verify Configuration** window will return the Administrator to the NatQuery desktop.

In the process of being returned to the NatQuery desktop, the use may see a message similar to Figure 10 (**Initialization Error**). This error should be considered normal, as at this point the Verify process may find that one or more DDM(s) do not have all of the related information needed. In response to this message, the user can click the **OK** button.

When the Administrator is back on the NatQuery desktop, the Administrator will now typically proceed to act upon the information seen in the Verify Configuration report as to what additional work need to be done to allow a given DDM to become Verified (I.E. address Occurrence Information for other DDMs and / or address Descriptor Statistic Information for any given DDM).

Descriptor Statistic Information

Descriptor Statistic Information provides NatQuery with the intelligence to understand the relative dynamics of each access path that may exist into a given file (Descriptor, Sub-Descriptor or Super-Descriptor). By capturing **Descriptor Statistics Information**, NatQuery is able to intelligently parse user-entered **Selection Logic** against the available I/O paths into a given file or set of files. This parsing gives NatQuery the ability to select which I/O path (ISN, Descriptor, Sub-Descriptor or Super-Descriptor) makes the most sense to use to resolve a given query, as well as additionally resolve the best statement (GET, HISTOGRAM, FIND, READ LOGICAL, READ BY ISN, or READ PHYSICAL) to use in conjunction with that access path.

For situations where NatQuery will be used as an End-User Query Tool, or will otherwise be used for *selective* data extraction against a given file, providing NatQuery with accurate Descriptor Statistic Information should be considered a requirement as it will be critical from a performance standpoint.

To assist the Administrator in capturing the required statistics, NatQuery contains a function which will generate Natural programs (specific to each DDM) that will obtain this information in an automated fashion. If the Administrator would like to take advantage of this function, the Administrator should follow the instructions below entitled Providing Descriptor Statistics Automatically.

For situations where NatQuery will be used as a Data Warehousing (DWH) Extract Tool or will otherwise be used for mass data extraction against a file, then providing NatQuery with accurate Descriptor Statistic Information should not be considered a requirement as DWH extraction efforts will in all likelihood require that files be read so as to extract all records. If NatQuery is being configured to support a DWH extraction effort and / or full file extraction is all that is desired against a specific file, then the Administrator should follow the instructions below entitled Providing Descriptor Statistics Manually.

With the sole exception of DDMs that correspond to Sequential Files, *every* DDM that is desired to be used as a data source for NatQuery must have some Descriptor Statistic Information entered in order to become usable for extraction. The issue then is whether or not the Descriptor Statistic information needs to be accurate (to allow for performance-oriented Selective Extraction) or not (to only support Mass Extraction).

Providing Descriptor Statistics Automatically

If NatQuery has been properly configured, it should now be able to generate the Natural program needed to capture the required Descriptor Statistics that will then support performance-sensitive selective data extraction. To proceed, it is assumed that NatQuery should have the ability to generate the required statistic-gathering program, it should have the ability to build the required JCL / Script template that will run this program, and it should be able to move this JCL / Script and program onto the Natural server platform (where in most cases it will be automatically executed).

Perform the following for each DDM:

1. **Invoke Administer Descriptor Statistics Function**
   On an empty NatQuery desktop, click on the **Administer** drop-down menu, and then click on **Environment Configuration / Descriptor Statistics**. These actions will invoke the **Administer Descriptor Statistics** window. Detailed help on the use of the **Administer Descriptor Statistics** window is available by clicking the **Help** button found on this window.

2. **Select DDM and Generate Descriptor-Statistics Processing**
   Using the list box that is associated with the text **File To Administer Descriptor Statistics for** in the upper-left corner, select the DDM that requires Descriptor Statistics. If the name of the DDM requiring Descriptor Statistics cannot be found in the combo box, then the DDM does not exist in the current Environment Configuration path. This may mean that the DDM was never imported, or that the current setting of the Environment Path needs to be changed.

   If the selected file requires an ADABAS password (this is not normally the case), make sure the box labeled **ADABAS password-protected** is checked. This will subsequently instruct NatQuery to prompt for an ADABAS password for this file whenever it is referenced for processing.

   As a result of selecting the file, a button in the upper right corner named **Generate** will become enabled and the Administrator should now click the **Generate** button. This action will cause NatQuery to internally generate a Natural statistics-gathering program that is specific to the named DDM.

   In response to clicking the **Generate** button, NatQuery will typically present a message box similar to Figure 6 (**FTP Request to Server**).

   Prior to continuing, the Administrator is cautioned that the program / request that has been generated may require a significant amount of time to execute successfully. The basic functioning of a generated Descriptor Statistics program is to perform a HISTOGRAM on each “key” field (Descriptors, Sub-Descriptors or Super-Descriptors) that exists in the source DDM. Therefore, depending upon the number of records in the source file and the number of keys, ***the processing required to complete this program may be substantial***. This fact should be considered prior to releasing a Descriptor Statistics gathering process, with a possible consideration being that this process may be best deferred to a different time for execution (like overnight). As a *general rule of thumb*, a Descriptor Statistics request will typically execute in less than 1 hour – however when a file has an exceedingly large number of records (for example substantially more than 2 million) and / or the file has a large number of "keys" (for example more than 5), the processing required to completely handle the given file may take more than 1 hour.

If the Administrator is comfortable in releasing the Descriptor Statistics gathering program, then in response to the message box shown in Figure 6 (**FTP Request to Server)** the **Yes** button can now be clicked. This will cause NatQuery to move the request to the Natural server environment, where in most cases it will automatically be executed.

Depending upon how NatQuery is being integrated to the server platform, the request may now execute immediately, or it may require intervention to be executed (in this case, it is suggested that this intervention be performed now).

If problems develop with the movement of the request to the server, then the **Troubleshooting** section should be referred to.

For initial executions of a Descriptor Statistic request, it is advisable that the Administrator manually monitor the execution of these requests on the server platform to insure that no problems are encountered with the **Production Request Process** template that was used to process this request. If any difficulties are encountered with the execution of the request then it is suggested that you refer to the section entitled **Troubleshooting**; for general questions on the operation of the Production Request Process template please refer to the section entitled **JCL / Script Template**.

If the request appears to have executed properly, then the next step may be performed.

3. **Check Server**
   Using the **Check Server** function (accessed by clicking the **Check Server** icon on the NatQuery toolbar), the user should be able to monitor the status of the Descriptor Statistic Request by using the **Check Server for Update** button while on the **Check Server** window.

   If after clicking the **Check Server for Update** button the request slot containing the Descriptor Statistic gathering request is updated to be "**DONE**", the results can be introduced into the workstation environment from the server environment by first clicking on any portion of the text that is associated with the completed Descriptor Statistics request, with this action then enabling the **Retrieve Request Output** button. Clicking the **Retrieve Request Output** button will cause NatQuery to begin the process of retrieving the Descriptor Statistics file onto the workstation. After the **Retrieve Request Output** button has been clicked, NatQuery will produce a window entitled **Save Extract Data File** that will allow the user to provide a custom name for the downloaded file, with NatQuery providing a default name of STATPROG.TXT. In virtually all cases this is an acceptable default, so the Administrator can simply click the **Save** button to proceed. In response to this action, NatQuery will typically respond with a message box that provides the final status of the download. The Administrator can click the **OK** button to close this window; this will return the user back to the **Check Server** window. The user can now click the **OK** button to close the **Check Server** window, which will return the user to the NatQuery desktop.

4. **Import Descriptor Statistics**
   With the Descriptor Statistics file downloaded, return to the **Descriptor Statistic Information** window by clicking on the **Administer** drop-down menu, then clicking **Environment Configuration \ Descriptor Statistics**. Using the list box associated with the text **File to Administer Descriptor Statistics for**, select the DDM that was previously selected in Step 2.

   NatQuery should respond by enabling the **Import** button. Clicking on the **Import** button will then populate the window with the Descriptor Statistic Information just downloaded. Click the **Apply Changes** button to save this information. NatQuery will respond by internally saving the Descriptor Information for the file just handled, and will then clear the **Administer Descriptor Statistics** window in anticipation of handling the Descriptor Statistics for another file.

   The Administrator can now close the **Administer Descriptor Statistics** window by clicking the **OK** button.

   As the **Administer Descriptor Statistics** window closes, NatQuery may respond by displaying a message box similar to the one seen in Figure 3 (**Verification Question**).

   It is suggested that the Administer click **Yes** to this prompt, and review the results of the **Verify Configuration** report as this report will guide the Administrator if any further action is necessary. Further information on the NatQuery **Verify Environment Configuration** function can be found in the section entitled Verifying an Environment Configuration.

   Closing the Verify Configuration window will return the Administrator to the NatQuery desktop.

   In the process of being returned to the NatQuery desktop, the use may see a message similar to Figure 10 (**Initialization Error**). This error should be considered normal, as at this point the Verify process may find that one or more DDM(s) do not have all of the related information needed. In response to this message, the user can click the **OK** button.

   When the Administrator is back on the NatQuery desktop, the Administrator will now typically proceed to act upon the information seen in the Verify Configuration report as to what additional work needs to be done to allow a given DDM to become Verified.

Providing Descriptor Statistics Manually

In situation where FTP integration is not configured, Descriptor Statistics can be provided manually. Perform the following:

1. **Invoke Administer Descriptor Statistics Function**
   On an empty NatQuery desktop, click on the **Administer** drop-down menu, and then click on **Environment Configuration / Descriptor Statistics**. These actions will invoke the **Administer Descriptor Statistics** window. Detailed Help on the use of the **Administer Descriptor Statistics** window is available by clicking the **Help** button found on this window.

2. **Select DDM and Manually Enter Descriptor-Statistics**
   Using the **Administer Descriptor Statistics** window, select a DDM that requires Descriptor Statistics. If the name of the DDM requiring Descriptor Statistics cannot be found in the combo box, then the DDM does not exist in the current Environment Configuration path. This may mean that the DDM was never imported, or that the current setting of the Environment Path needs to be changed.

   If the Administrator is referring to this step, then either the use of NatQuery will be to support full file extraction (where accurate Descriptor Statistics are not needed), or the Administrator otherwise desires to enter Descriptor Statistics manually as opposed to obtaining these automatically as described above.

   If Descriptor Statistics are NOT needed for a given file (such as is typically the case for **DWH** efforts), then the Administrator should enter the approximate number of records that exist in the file into the field named **Total Number of Records in File**, and then click the button named **Zero All**. Clicking this button will zero out all I/O path entries, which will have the effect of forcing NatQuery to only ever be able to generate a Read Physical against the specific file. The Administrator should then click the **Apply Changes** button, and then click the **OK** button to close the **Administer Descriptor Statistics** window. If this paragraph has been followed, then the user can now move to step 4.1.3 below.

   If Descriptor Statistics will be entered manually, then for each I/O path into the file (Descriptor, Super-Descriptor, Sub-Descriptor) that NatQuery should consider, the manual entry of 4 pieces of information will be required: The **Average** number of records returned for any given unique value, the **Maximum** number of records that may exist for any given value, the **Total** number of records for all values, as well as the **Count** of unique values. The Administrator would begin by entering the approximate number of records that exist in the file into the field named **Total Number of Records in File**, and then proceed to enter the required information for each Descriptor as well as for each component level of Super-Descriptors. Using the graphical controls at the lower portion of the window, the Administrator will selectively enter the required numbers for each entry, then click the **Apply** button. This process is then repeated for each I/O path that NatQuery should consider (possibly using the **Zero Entry** button to zero-out I/O paths that should not be considered by NatQuery), until the **Apply Changes** button becomes available. Clicking the **Apply Changes** button will cause NatQuery to internally save the Descriptor Statistic information just entered, and it will then clear the **Administer Descriptor Statistics** window in anticipation of handling the Descriptor Statistics for another file.

The Administrator can now close the **Administer Descriptor Statistics** window by clicking the **OK** button.

As the **Administer Descriptor Statistics** window closes, NatQuery may respond by displaying a message box similar to the one seen in Figure 3 (**Verification Question**).

It is suggested that the Administer click **Yes** to this prompt, and review the results of the **Verify Configuration** report as this report will guide the Administrator if any further action is necessary. Further information on the NatQuery **Verify Environment Configuration** function can be found in the section entitled Verifying an Environment Configuration.

Closing the Verify Configuration window will return the Administrator to the NatQuery desktop.

In the process of being returned to the NatQuery desktop, the user may see a message similar to Figure 10 (**Initialization Error**). This error should be considered normal, as at this point the Verify process may find one or more DDM(s) do not have all of the related information needed. In response to this message, the user can click the **OK** button.

When the Administrator is back on the NatQuery desktop, the Administrator will now typically proceed to act upon the information seen in the Verify Configuration report as to what additional work need to be done to allow a given DDM to become Verified.

File Relationship Information

File Relationship information must be provided if there is any requirement that any single extract should be able to access more than one physical file. If only a single DDM has been downloaded into NatQuery then this step can be skipped as there must be at least two DDMs available to define a File Relationship.

In some cases, for example when NatQuery is used to support Data Warehousing (DWH) efforts where it is somewhat typical that only single-file extracts are required, File Relationship information may not be needed. If the Administrator is convinced that only single-file extracts will be needed, and no automatic linking of multiple files are required for any extracts, then the Administrator may jump to step 6.

Above and beyond the standard types of file relationship cardinality supported by PREDICT, NatQuery allows for one further file relationship; that of a "Lookup". A Lookup file relationship is defined as always being a 1:1 relationship, and from an End-user perspective a Lookup relationship gives the appearance that the field(s) designated by a Lookup relationship are directly "attached" to a given file. An example of this is a situation where a “code” exists as a field on one file and that code can be programmatically translated into a text string by accessing a separate "Lookup" or “Code Reference” file. From the End-user's perspective, when the base file is selected for data extraction, this base file will show the "lookup" translation text field as if the translation text field was part of the base file – thereby allowing the translation to be automatic and not require further End-user involvement.

The process of defining a File Relationship essentially involves two things: It involves the identification of a Descriptor or Super-Descriptor that supports the required linking, and it additionally involves designating how the component(s) of the Descriptor or Super-Descriptor is properly initialized.

With this information made available, NatQuery can then automatically support allowing multiple logically-related files to be selected so that data fields can be retrieved from some or all of these files.

To build File Relationship information, perform the following:

1. **Invoke Administer File Relationships**
   On the **Administer** drop-down menu, click on **Environment Configuration**, and then click on **Administer File Relationships**. This will invoke the **Administer File Relationships** window.

2. **Build Required File Relationships**
   Using the **Administer File Relationships** window, build all required inter-file relationships. Detailed help on how to use the **Administer File Relationships** window can be found by pressing the **Help** button while in this function.

   To enter File Relationship information, the Administrator will begin by selecting a file (**File**

#1) that should support an automatic link to another file, and the Administrator will then select this second file (**File #2**).

If the name of either end of a file link cannot be found in the **File #1** or **File #2** combo boxes, then the corresponding DDM does not exist in the current Environment Configuration path. This may mean that this DDM was never imported, or that the current setting of the Environment Path needs to be changed.

With **File #2** selected, the “key” (Descriptor, Super-Descriptor or Sub-Descriptor) that will physically support the linking to File #2 is selected using the **File 2 Key Field** list box. Once the **File 2 Key Field** is selected, the Administrator will then “paint” how each component of the key gets initialized, and will then further define the cardinality of this relationship.

When the definition of a needed File Relationship has been completed, the **Apply** button will become enabled, and clicking the **Apply** button will cause NatQuery to internally save this File Relationship and then clear the windows graphical controls in anticipation of handling another File Relationship.

After all needed File Relationships have been entered, the Administrator will click the **OK** button to close the Administer File Relationship window.

At any time that the Administer File Relationships window is closed following the modification or entry of File Relationship information for any given file, NatQuery will respond to this action by displaying a message similar to Figure 3 (**Verification Question**).

It is suggested that the Administer click **Yes** to this prompt, and review the results of the **Verify Configuration** report as this report will guide the Administrator if any further action may be necessary. Further information on the NatQuery **Verify Environment Configuration** function can be found in the section entitled Verifying an Environment Configuration.

Closing the **Verify Configuration** window will return the Administrator to the NatQuery desktop.

In the process of being returned to the NatQuery desktop, the Administrator may see a message similar to Figure 10 (**Initialization Error**). This error should be considered normal, as at this point the Verify process may find that one or more DDM(s) do not have all of the related information needed. In response to this message, the user can click the OK button.

When the Administrator is back on the NatQuery desktop, the Administrator will now typically proceed to act upon any deficiencies seen in the Verify Configuration report as to what additional work may be needed to allow a given DDM to become Verified.

Field Sign Byte Information

Sign Byte Information pertains to how NatQuery will handle numeric fields for which a sign byte must be provided when the specific numeric field is extracted.

By default, NatQuery assumes that all numeric fields are positive: This is done to minimize the possible size of a given extract record as much as possible. Under this assumption, NatQuery will *not* automatically include a sign byte for any numeric field when the field is extracted.

If **Sign Byte Information** is required of any numeric field being output, then NatQuery needs to be informed of this. Once this information is provided, NatQuery will automatically provide a sign position whenever the designated field(s) is selected for output.

If there are no numeric fields in a file that requires a sign position when extracted, then the Administrator may ignore providing this information for the given DDM. If however there are one or more fields in a file that does require a sign position, these fields must be designated to NatQuery.

Please note that Sign Byte information for a given field may also be designated through the use of Field Edit Masks; for further information on Field Edit Mask please refer to the section entitled **Field Edit Mask Information**.

To provide **Sign Byte Information** for any field, perform the following:

1. **Invoke Sign Byte Information Function**
   Click on the Administer drop-down menu, then click on Environment Configuration, and then click on Sign Byte Information. This will invoke **Administer Sign Byte Information** window.

2. **Enter Required Sign Byte Information**
   On the top of the **Sign Byte Information** window, there is a drop down list box named **File to Administer Sign Byte Information for**. Using this list box, select the file that contains the field or fields that will require a sign byte when they are extracted. Selecting a given file will populate the **Sign Byte Information** grid with just those fields that are numeric in nature.

   If the name of a DDM requiring Sign Byte Information cannot be found in the list box, then the corresponding DDM does not exist in the current Environment Configuration path. This may mean that this DDM was never imported, or that the current setting of the Environment Path needs to be changed.

   Using the grid control that displays these numeric fields, navigate to the field or fields that require a sign byte. A field of interest may be double-clicked to toggle between a sign byte being required or not; or alternatively the field can be single-clicked with the **Sign Byte** list box control being used to accomplish the toggle.

When all fields that require a sign byte are recorded, click the **Apply Changes** button. This action will record the entered sign byte information into NatQuery, and will clear the **Administer Sign Byte Information** window in anticipation of handling another file. You may then continue with the next file that may require numeric fields to have sign byte information and repeat the above procedure.

3. **Close Sign Byte Information Function**
   When all fields in all required files are appropriately handled, you may then click on the **OK** button to close the **Administer Sign Byte Information** window.

   Making changes to **Sign Byte Information** does not, in and of itself, require that an Environment Configuration be Verified.

Field Edit Mask Information

Edit Mask Information pertains to how NatQuery handles the outputting of designated DDM fields.

By default, NatQuery will assume that all fields being output will be converted to Alpha-numeric equivalents. For source fields that are numeric in nature, this means that when the NatQuery generates the logic to handle the output of a given source field, an edit mask will be applied automatically to the field’s output value.

NatQuery’s default handling of ADABAS Source fields is as follows:

- Default Handling for Date & Time Fields,
- Default Handling for Numeric Fields, and
- Default Handling for Binary Fields

Using functions of NatQuery, it is possible to over-ride specific default handling of fields, according to the following hierarchy:

- Over-riding Field Output Handling by Administrative Sign Byte Information,
- Over-riding Field Output Handling by User-Requested Leading Zero Suppression, and
- Over-riding All Field Output Handling by Providing Edit Masks

Default Handling for Date & Time Fields

For Date and Time fields, NatQuery applies the following Edit Masks by default:

Field Format	Output Format/Length	Edit Mask Applied
D	A10	Specified by Date Format, which should conform to the setting of the NATPARM DTFORM. Example: EM=MM/DD/YYYY
T	A19	Specified by the setting of Date Format, followed by a space, followed by Time. Example: EM=MM/DD/YYYY' 'HH:II:SS

If the Administrator wishes to change the output for any specific D or T field to be another format, for example the ability to add tenths of seconds, then this can be done using the Edit Mask function as described further below.

Default Handling for Numeric Fields

For Numeric fields, NatQuery applies the following Edit Mask Handling by default. Note that NatQuery is separately sensitive to the Decimal Character in use at a given site.

Field Format	Output Format/Length	Edit Mask Applied
N.3	A4	Example: EM=.9(3) or EM=,9(3)
N5.2	A8	Example: EM=9(5).9(2) or EM=9(5),9(2)
N7	A7	Example: EM=9(7)
P.3	A4	Example: EM=.9(3) or EM=,9(3)
P5.2	A8	Example: EM=9(5).9(2) or EM=9(5),9(2)
P7	A7	Example: EM=9(7)
I1	A4	Example: EM=9(4)
I2	A6	Example: EM=9(6)
I4	A11	Example: EM=9(11)

Default Handling for Binary Fields

For Binary fields, NatQuery applies the following handling by default.

Field Format	Output Format/Length	Edit Mask Applied
B1	N3	EM not needed
B2	N5	EM not needed
B3	N8	EM not needed
B4	N10	EM not needed

Over-riding Field Output Handling by Administrative Sign Byte Information

For just the Numeric field category, the default handling of fields being output will be modified through the Administrative use of the **Sign Byte Information** function, with the Sign Byte information function dictating whether or not a Numeric field needs to have a Sign Byte output with the field's value. Information on assigning a Sign Byte to a numeric field is described in the section entitled Field Sign Byte Information.

If a Numeric field has a Sign Byte assigned by the Administrator, then the field will be handled as demonstrated in the following table. Not the increase in Output Length and the use of the "S" to handle the Sign position.

Field Format	**Output Format/Length**	**Edit Mask Applied**
N.3	A5	Example: EM=S.9(3) or EM=S,9(3)
N5.2	A9	Example: EM=S9(5).9(2) or EM=S9(5),9(2)
N7	A8	Example: EM=S9(7)
P.3	A5	Example: EM=S.9(3) or EM=S,9(3)
P5.2	A9	Example: EM=S9(5).9(2) or EM=S9(5),9(2)
P7	A8	Example: EM=S9(7)
I1	A5	Example: EM=S9(4)
I2	A7	Example: EM=S9(6)
I4	A12	Example: EM=S9(11)

Over-riding Field Output Handling by User-Requested Leading Zero Suppression

When a user creates an extract process, they are given the option to **Suppress Leading Zeros in Numeric Output**.

If the user utilizes this option, then all numeric values referenced in that specific extract will be output with leading zeros suppressed. This is accomplished through the use of the “Z” Edit Mask character, and will be sensitive to the use of any Administratively assigned Sign Byte information.

Over-riding All Field Output Handling by Providing Edit Masks

To permanently change the behavior of how any given field gets output, the Administrator can assign any given field a custom Edit Mask – with this assignment over-riding the default assignment of the field, any assigned Sign Byte information AND any user-dictated Suppression of Leading Zeros (for numeric fields).

To access the Edit Mask function, the Administrator would perform the following:

1. **Invoke Edit Mask Function**
   Click on the **Administer** drop-down menu, then click on **Environment Configuration**, and then click on **Edit Masks**. This will invoke the **Administer Edit Masks** function. Specific help on using the **Administer Edit Mask** function is available by clicking the **Help** button while in this function.

2. **Enter Required Edit Mask Information**
   On the top of the **Administer Edit Mask**s window, there is a drop down list box named **Select File to Administer Edit Mask Information for**. Using this list box, select the file that contains a field that requires an Administratively assigned Edit Mask when it is designated to be extracted. Selecting a given file will populate the **Custom Edit Mask** grid with the fields available from the selected DMM.

   Using the grid control that displays these numeric fields, navigate to the field or fields that require an **Edit Mask**. The field that requires an **Edit Mask** should be selected by clicking on the field's name – this action will populate the graphical controls that exist below the grid. Using these controls the Administrator will enter the new **Alpha Length** that will contain the Edit Masked output of the field, as well as entering the **Edit Mask** value that will be used to populate the new definition. Entering both an **Alpha Length** and an **Edit Mask** for a given field will enable the Apply button; clicking the Apply button will move the new values for **Alpha Length** and **Edit Mask** into the grid.

   When entering the **Alpha Length** and **Edit Mask** values, the onus is on the Administrator to insure that the **Alpha Length** is correct, and that the **Edit Mask** value is correct: In current versions of NatQuery there is no editing done on this.

   When all fields that require an **Edit Mask** are recorded, click the **Apply Changes** button. This action will record the entered **Edit Mask** information into NatQuery, and will clear the **Administer Edit Masks** window in anticipation of handling another DDM. You may then continue with the next file that may require **Edit Masks** and repeat the above procedure.

3. **Close Administer Edit Masks Function**
   When all fields in all required files are appropriately handled, you may then click on the **OK** button to close the **Administer Edit Masks** window.

   Making changes to **Edit Masks** does not, in and of itself, require that an Environment Configuration be Verified.

Redefine DDM Fields

Quite often, a given DDM may contain one or more “complex fields” or fields which are actually comprised of components or redefinitions.

While an End-User has the ability to redefine DDM source fields through the use of user-defined Query variables, a user-built Redefine Variable is exclusive to the specific query it is built into. This limitation could therefore cause the user to redundantly create user-defined redefinition variable each time a query is built that needs to extract a specific redefinition. Beyond this, and to shield a user from as much complexity as possible, NatQuery limits an End-User to only redefine a given DDM field into Alphanumeric components (fields with a format of “A”) which may not accurately address the imbedded component’s true format definition).

To address these issues, the Administrator is given the capability of redefining any given field at the DDM level, with each redefined component being assigned any valid Format / Length. By providing this capability, an Administrator can create redefined components that become immediately available to a user when the user selects a DDM that contains an Administratively-defined Redefinition(s). Additionally, each component can be given a Format that is appropriate to the component’s true use.

Beyond the ability to administratively redefine a given DDM field, depending on the result, the Administrator should also be aware of the ability to provide File Variables, with File Variables being presented in the section entitled File Variables.

The ability to provide a DDM Redefinition is a sub-function of the **Build / Edit DDM** function. To provide an Administratively-provided field redefinition, the Administrator would perform the following:

1. **Invoke Build / Edit DDM Function**
   Start NatQuery if not already started. On an empty NatQuery desktop, click on the **Administer** drop-down menu, and then click on **Environment Configuration** / **DDMs & FDTs** / **Build & Edit DDM**. This will invoke the **Build / Edit DDM** function. Specific help on using the **Build / Edit DDM** function is available by clicking the **Help** button while in this function.

2. **Select DDM to have Redefinitions Applied**
   On the top of the **Build / Edit DDM** window, there is a drop down list box named **Select DDM**. Using this list box, select the DDM that contains the field or fields that will require an Administrative Redefinition. Selecting a given file will populate the list box in the center of the **Build / Edit DDM** window with the information from the DDM.

   For each DDM field that requires an Administrative redefinition, perform the following:

   2.1. **Select Field to be Redefined**
   Using the list box in the center of the window, the Administrator will locate and select a field that requires an Administrative redefinition. Selecting a given field will populate

the graphical controls in the lower portion of the window, and will additionally enable a button located in the lower right named **Redefine**.

The Administrator should therefore click the **Redefine** button.

2.2. **Redefine DDM Field window**
Using the controls on the **Redefine DDM Field** window, the Administrator can proceed to enter the appropriate field redefinition. Detailed help on using the **Redefine DDM Field** window can be found by clicking the **Help** button while in this function.

To enter a redefinition for a DDM field, the Administrator will provide a **Field Name** (this will be the name of a redefinition component that the user will see), as well as the **Format** and **Length** of each component of the base field. Once the **Field Name**, **Format** and **Length** of a component are entered, the Administrator will click the **Replace** button – which will transfer the component definition into the central grid. When this occurs, a new grid entry will be automatically built and selected, such that the Administrator can immediately enter the **Field Name**, **Format** and **Length** fields and then hit **Replace** to build the next component. This process continues until all components of the base field are defined.

Additional functions of the **Redefine DDM Field** window allow a new component definition to be inserted (**Insert** button), a selected component to be deleted (**Delete** button), or all existing components to be deleted (**Delete All** button).

When all redefined components of the base selected field are accounted for, the Administrator will then click the **OK** button to return to the **Build / Edit DDM** window.

In returning to the **Build / Edit DDM** window, the Administrator will see that the redefinition has been inserted into the DDM using syntax that NatQuery can then suitably interpret.

If other fields require a DDM redefinition at this time, then this can be accomplished using the procedure just outlined.

Note:
Deleting all components of an existing redefined base field and then clicking the **OK** button while on the **Redefine DDM Field** window will have the effect of completely removing any base field redefinition from the DDM.

3. **Save DDM**
When all required administrative field redefinitions have been handled and are seen on the **Build / Edit DDM** window, the Administrator should click the **Save DDM** button. Clicking the **Save DDM** button will instruct NatQuery to save the new definition of the DDM into the Environment Configuration, and NatQuery will then remind the Administrator that the Environment must be Verified prior to the changed DDM becoming available for extraction handling.

Subsequent to the Save DDM button being pressed, the Administrator will be returned to the Build / Edit DDM window. If other DDMs require administrative definitions, these additional DDMs may be handled using the procedure described above. Once all needed DDM fields have been redefined, the Administrator can close the Build / Edit DDM window by clicking the OK button.

As the Build / Edit DDM window closes, NatQuery will detect that one or more DDMs have been changed, and will present a message box similar to that seen in Figure 3 (**Verification Question**).

It is suggested that the Administer click **Yes** to this prompt, and review the results of the **Verify Configuration** report as this report will guide the Administrator if any further action may be necessary. Further information on the **NatQuery Verify Environment Configuration** function can be found in the section entitled Verifying an Environment Configuration.

Closing the **Verify Configuration** window will return the Administrator to the NatQuery desktop.

In the process of being returned to the NatQuery desktop, the Administrator may see a message similar to Figure 10 (**Initialization Error**). This error should be considered normal, as at this point the Verify process may find that one or more DDM(s) do not have all of the related information needed. In response to this message, the user can click the **OK** button.

When the Administrator is back on the NatQuery desktop, the Administrator will now typically proceed to act upon any deficiencies seen in the Verify Configuration report as to what additional work may be needed to allow a given DDM to become Verified.

File Variables

When handling data extraction, NatQuery provides a user with the ability to create user-defined variables, with these variables extending what can be extracted beyond the source fields themselves. When using user-defined variables however, these variables are specific to a given query, meaning that if the same variable is needed across several extracts, the user could be forced to redundantly define variables into each query where they may be needed.

To address this potential limitation, the Administrator is given the ability to pre-define variables against a given source file, such that these variables are selectable by the user as if they were part of a source file's DDM. By pre-defining variables, the administrator can save the user's time when these users have the repetitive need to constantly define the same variables over and over again to obtain specific values.

NatQuery supports several different types of variable definitions, and all of these types can be administratively pre-defined as File Variables. The variables that can be defined are:

- Redefinition
- Constant
- Expression
- Dynamic
- Compress

When considering the use of a File Variable, the Administrator should also be aware that the use of a Global Variables or possible Redefine DDM Fields may also achieve the desired goal.

File Variables differ from **Global Variables** in that **File Variables** will be associated only with the file they are defined against, whereas **Global Variables** will be associated with all source files. Additionally, a **File Variable** cannot be referenced within a **Global Variable**, however they can be referenced in other **File Variables** for the same file (Administratively defined), any **Query Variables** (defined by an extract user) or used directly as an extract field. Further information on **Global Variables** can be found in the section entitled Global Variables.

Creating, modifying, or deleting a **File Variable** generally uses the same variable handling functions that allow a user to define a query variable. Adding, modifying, or deleting a **File Variable** does not require that a Verify Environment Configuration be performed.

To create a **File Variable**, perform the following:

1. **Invoke Administer Pre-Defined File Variable Function**
   Start NatQuery if not already started. On an empty NatQuery desktop, click on the **Administer** drop-down menu, and then click on **Environment Configuration / File Variables**. This will invoke the **Administer Pre-Defined File Variables** function. Detailed help on using the **Administer Pre-Defined File Variables** function can be found by clicking the **Help** button while in this function.

2. **Select the File That Requires File Variable Manipulation**
   Using the drop-down combo box, locate and then select the file that requires a **File Variable** by clicking on its name. This action will enable the button labeled **Pre-Define Variable for Selection**.

   Click the **Pre-Define Variable for Selection** button; this action will invoke the **List Pre-Defined File Variable** window.

   To close the **Administer Pre-Defined File Variable** function, the user will click the **OK** button. This action will return the user to the NatQuery desktop.

3. **List Pre-Defined File Variable Window**
   When the **List Pre-Defined File Variable** window is presented, it will display any pre-existing **Global Variables** that have been previously pre-defined. Subsequent to displaying any **Global Variables**, any pre-existing **File Variables** for the selected file will be displayed.

   To add a new **File Variable**, the Administrator will click the **Add** button. This will invoke the **Define File Variable** window, and the user can then continue with step 4.

   To modify the definition of an existing **File Variable**, the Administrator will begin by locating the variable needing to be modified, select that variable by single-clicking it's name, and then clicking the **Modify** button. This will invoke the **Define Variable** window, and the user can then continue with step 4.

   To delete an existing **File Variable**, the Administrator will locate the variable needing to be deleted, select that variable by single-clicking its name, and then click the **Delete** button. NatQuery will respond to this by prompting the user for confirmation of the delete action. Clicking **Yes** to this prompt will cause NatQuery to internally delete the **File Variable**'s definition; clicking **No** will forego the delete operation. Either way, the Administrator will subsequently be returned to the **List Pre-Defined File Variables** window.

   To close the **List Pre-defined File Variable** window, the user will click the **OK** button. This action will return the user to the **Administer Pre-Defined File Variable** function, and the user may then refer to step 2 for how to proceed.

4. **Define File Variable Window**
   The **Define File Variable** window allows for the creation of a new **File Variable** and also allows the modification of an existing **File Variable**. Detailed information on how to use the **Define File Variable** is available by clicking the **Help** button while on this window.

   Using the controls on the **Define File Variable** window, the Administrator will either add the definition of a new **File Variable** and / or will modify the definition of an existing **File Variable**.

   When the addition or modification is complete, the user will click the **OK** button to close the **Define File Variable** window. This action will return the user to the **List Pre-Defined File**

Variables window. The user may then refer to step 3 for how to proceed.

DDM Editor

To assist an Administrator in the creation and modification of DDMs, NatQuery provides a **DDM Editor** function that handles these tasks. Additionally, the NatQuery **DDM Editor** allows for the creation, modification and deletion of Administratively-defined **Redefined DDM Fields**.

If the intent of the Administrator is to handle the creation, modification, or deletion of a Redefine DDM field, then the section entitled Redefine DDM Fields should now be referred to.

While the **DDM Editor** can be used to create a new DDM or modify an existing DDM, the DDM Editor will typically only be used in the following situations:

- A DDM relating to a **Sequential** File needs to be created or modified, and no DDM for this Sequential File exists anywhere else; or

- One or more fields in any DDM require the creation, modification or deletion of a **Redefine DDM Field**.

In all other situations, it would generally be expected that a version of the needed DDM exists on the Natural platform against which NatQuery is expected to operate. Therefore, as opposed to creating or modifying a DDM "by hand" using an editor, it is typically preferable to download any needed DDMs and then **Import** them into NatQuery (either a completely new DDM or a newer version of an existing DDM). Information on importing a DDM can be found in the section entitled Adding a New DDM, or alternatively the section entitled Modifying An Existing DDM.

If the intent is to use the DDM Editor to create or modify a **Sequential** File, or to proceed with editing against a DDM that does not represent a **Sequential** File for whatever reason, then the following steps would be performed:

1. **Invoke Build / Edit DDM Function**
   Start NatQuery if not already started. On an empty NatQuery desktop, click on the **Administer** drop-down menu, and then click on **Environment Configuration** / **DDMs / FDTs** / **Build / Edit DDM**. This will invoke the **Build / Edit DDM** function.

   Detailed help on using the **Build / Edit DDM** function can be found by clicking the **Help** button while in this function.

   In general, the DDM Editor works by providing text boxes for the various DDM Header fields. Beyond this, the "guts" of the DDM are displayed in the central textbox of the Edit DDM window. Individual lines in this textbox are changed by selecting them, and then utilizing the text boxes in the lower portion of the screen in conjunction with the various function buttons that will become available.

2. **Select DDM to Handle**
   At the top of the **Build / Edit DDM** window there is a combo box entitled **Select DDM**.

If the need is to create a new DDM, then the first choice presented in the Select DDM combo box should be selected, which is the selection named **<Add-New-DDM>.**

If the need is to Modify an existing DDM, then the name of this DDM should be located in the **Select DDM** combo box and selected. If the name of the DDM to be changed cannot be located in the **Select DDM** combo box, then this DDM does not exist in the current Environment Configuration path. Likely causes for this situation may be that the DDM was never physically Imported into the current Environment Configuration, or the Environment Configuration is pointing at an incorrect path.

3. **Handle DDM Header Editing (Add-New-DDM)**
   If a new DDM is being created, then the administrator will enter the "header" information for the DDM. If an existing DDM is being edited, then the header information will have already been entered, and in current versions of NatQuery this Header information cannot be changed, so the reader can proceed to the next step.

   Entering DDM Header information will begin by entering the name of the DDM into the text field called **DDM Name**.

   If the DDM being created will refer to a **Sequential** file, then the checkbox labeled **Sequential** file will be checked by default. With the **Sequential** checkbox checked, the text fields labeled **DBID** and **FNR** will be disabled with both of these fields being assigned a text value of "SEQ".

   If the DDM being created will refer to a file that does not represent a **Sequential** file, then this will be indicated by un-checking the **Sequential** file checkbox. With the Sequential file checkbox un-checked, the text fields entitled **DBID** and **FNR** will become enabled. For a non-Sequential file, a correct numeric value for **DBID** may be entered (but is optional) and a correct numeric value for **FNR** must be entered.

   When creating **Sequential** files, it is usually desirable to "link" the DDM to the name of the physical file that contains the sequential data. NatQuery supports this linking by allowing an Administrator to specify the physical name as a comment in the DDM as the first line. If this comment line is entered then NatQuery will automatically provide this file name in JCL or Script as appropriate, thus removing the need for a NatQuery user to have to remember this information. To take advantage of this feature, the Administrator will enter a comment line as the first entry line of the DDM, and this will be accomplished by placing an asterisk ("*") in the "**T**" (Type) column. The **Text** field should then be filled with the actual name of the Sequential file, along with a **Remark** entry of "**<< DDM DATASET NAME**".

4. **Handle DDM Line Editing**
   To **add** a new line into a DDM, the line in the DDM that will be added will be selected in the central text box; this action will enable the line of text boxes in the lower portion of the screen. These text boxes allow for the entry of the individual fields that comprise a

DDM line entry and represent an entry's **Type** (labeled "**T**"), **Level** (labeled "**L**"), **Two-Character Database Name** (labeled "**DB**"), **Field Name** (labeled "**NAME**"), **Format** (labeled "**F**"), **Length** (labeled "**LENG**"), **Suppression** (labeled "**S**"), **Descriptor** flag (labeled "**D**") and **Remarks ("REMARKS)**. Once the appropriate values for these fields have been entered, the user will click the **Replace** button, with this action placing the contents of the separate fields into the single highlighted line in the central textbox.

To **insert** a line into the central text box, the line preceding the point of insertion would be selected; this action will enable the **Insert** button. Clicking the **Insert** button will insert a blank line into the central textbox immediately following the highlighted line, with the just inserted line then becoming selected. The focus would then shift to the individual DDM line entry text fields which will then allow entry for the DDM line entry. Once the appropriate values for the field have been entered, the user will click the **Replace** button, with this action placing the contents of the separate fields into the single highlighted line in the central textbox.

To **modify** an existing line, the line would be selected in the central text box; this action will populate the line's contents into individual text boxes in the lower portion of the window where they can be changed as needed. Once all changes to the line have been made, the user will click the **Replace** button to put the contents of the separate fields into the highlighted line in the central textbox.

To **delete** an existing line, the line would be selected in the central text box; this action will enable the **Delete** button. Clicking the **Delete** button will remove the highlighted line from the central textbox.

To **delete all** lines in the central text box, the **Delete All** button should be clicked. Clicking the Delete All button will result in a message box being displayed that will seek confirmation of the Delete All action. If such confirmation is given, all lines in the central textbox will be removed.

To **cancel** at any time, the **Cancel** button would be clicked. Clicking **Cancel** will discard any / all changes just made to the current DDM, and will return the user to the NatQuery desktop.

To **save** any / all changes made, the **Save DDM** button would be clicked. Clicking the **Save DDM** button will (after editing) save any / all changes made to the DDM, and will then reset the **Build / Edit DDM** window for possible further DDM editing.

When all changes have been made, the administrator will click the **OK** button to close the **Build / Edit DDM** window; this action will return the user to the NatQuery desktop.

If a previously verified DDM has been changed, the act of closing the DDM Editor will cause NatQuery to produce a message similar to that seen in Figure 3 (**Verification Question**). When presented with this message, it is advisable to indicate that Verification should be performed with the resulting report examined for any possible errors on the affected DDM(s) for further possible Administrative action.

Global Variables

When handling data extraction, NatQuery provides a user with the ability to create user-defined variables, with these variables extending what can be extracted beyond just available source fields themselves. When using user-defined variables however, these variables are specific to a given query, meaning that if the same variable is needed across several extracts, the user may be forced to redundantly define variables into each query where they may be needed.

To address this potential limitation, the Administrator is given the ability to pre-define **Global Variables**, such that these variables are selectable by the user as if they were part of any source file. By pre-defining variables, the administrator can save the user's time when these users have the repetitive need to consistently define the same variable over and over, and the Administrator can additionally save his / her own time by creating variables that can then be used in the definition of individual **File Variables**.

File Variables differ from **Global Variables** in that **File Variables** will be associated only with the file they are defined against, whereas **Global Variables** will be associated with all source files. Additionally, **Global Variables** can be referenced in the definition of other **Global Variables**, any **File Variables** (Administratively defined), any **Query Variables** (defined by an extract user) or used directly as an extracted field (as selected by a user in a query).

NatQuery supports several different types of variable definitions, however due to the nature of how **Global Variables** operate, only the following types of variables can be administratively pre-defined as **Global Variables**:

- Constant
- Expression
- Dynamic

When considering the use of a **Global Variable**, the Administrator should also be aware that the use of File Variables or the possible us of Redefine DDM Fields may be used to achieve the desired goal(s). Further information on File Variables can be found in the section File Variables, with further information on Redefining DDM Fields can be found in the section Redefine DDM Fields.

Creating, modifying, or deleting a **Global Variable** generally uses the same variable handling functions that allow a user to define a query variable. Adding, modifying, or deleting a **Global Variable** does not require that a Verify Environment Configuration be performed.

To create a **Global Variable**, perform the following:

1. **Invoke Administer Pre-Defined File Variable Function**
   Start NatQuery if not already started. On an empty NatQuery desktop, click on the **Administer** drop-down menu, and then click on **Environment Configuration / File Variables**. This will invoke the **Administer Pre-Defined File Variables** function. Detailed help on using the **Administer Pre-Defined File Variables** function can be found by clicking

the **Help** button while in this function.

2. **Select <Global Variable>**
   Using the drop-down combo box, locate and then select the tag labeled **<Global Variable>**. This action will enable the button labeled **Pre-Define Variable for Selection**.

   Click the **Pre-Define Variable for Selection** button; this action will invoke the **List Pre-Defined Global Variable** window, and the user can proceed with step 3.

   To close the **Administer Pre-Defined Global Variable** function, the user will click the **OK** button. This action will return the user to the NatQuery desktop.

3. **List Pre-Defined Global Variable Window**
   When the **List Pre-Defined Global Variable** window is presented, it will display any pre-existing **Global Variables** that have been previously pre-defined.

   To add a new **Global Variable**, the administrator will click the **Add** button. This will invoke the **Define Global Variable** window, and the user can then continue with step 4.

   To modify the definition of an existing **Global Variable**, the administrator will begin by locating the variable needing to be modified, select that variable by single-clicking it's name, and then clicking the **Modify** button. This will invoke the **Define Global Variable** window, and the user can then continue with step 4.

   To delete an existing **Global Variable**, the administrator will locate the variable needing to be deleted, select that variable by single-clicking its name, and then click the **Delete** button. NatQuery will respond to this by prompting the user for confirmation of the delete action. Clicking **Yes** to this prompt will cause NatQuery to internally delete the **Global Variable**'s definition; clicking **No** will forego the delete operation. Either way, the Administrator will subsequently be returned to the **List Pre-Defined Global Variables** window.

   To close the **List Pre-defined Global Variable** window, the user will click the **OK** button. This action will return the user to the **Administer Pre-Defined File Variable** function, and the user may then refer to step 2 above on how to proceed.

4. **Define Global Variable Window**
   The **Define Global Variable** window allows for the creation of a new **Global Variable** and also allows for the modification of an existing **Global Variable**. Detailed information on how to use the **Define Global Variable** is available by clicking the **Help** button while on this window.

   Using the controls on the **Define Global Variable** window, the Administrator will either add the definition of a new **Global Variable** or modify the definition of an existing **Global Variable**.

   When the addition or modification is complete, the user will click the **OK** button to close the

Define Global Variable window. This action will return the user to the **List Pre-Defined Global Variables** window. The user may then refer to step 3 for how to proceed.

JCL / Script Templates

The execution of all NatQuery generated processes occur within a batch environment on the Natural server platform.

To support this execution, NatQuery handles the generation of either the Job Control Language (JCL) or Scripts that then enable a given process to execute.

NatQuery JCL / Script templates are generic in nature, with the specific attributes / references that will make the generic template unique to a specific task being referenced through pre-defined **Dynamic Substitution Variables**. A typical JCL / Script template will always contain one or more references to **Dynamic Substitution Variables**, with these variables being identifiable with a prefix of "&&" (ampersand, ampersand), followed by a text string that describes the use of a substitution value. For example, the dynamic substitution variable &&USER-ID will be substituted with the actual value of the submitting user's User ID.

Just prior to a given request being submitted by NatQuery to the Natural / ADABAS server platform, NatQuery will select a specific template that will support the requested task. Once the template is selected, NatQuery will scan the selected template for the existence of any **Dynamic Substitution Variables**. If any **Dynamic Substitution Variables** are found, these variables will be physically substituted with appropriate replacement values, resulting in a fully executable JCL stream or Script.

For a complete reference of the name and description of the dynamic variables that NatQuery JCL / Script templates may utilize, please refer to the section entitled NatQuery Dynamic Substitution Variable Reference Table.

In approaching the creation of viable JCL / Script templates, NatWorks wishes to make it clear that the configuration of JCL / Scripts is the most difficult part of implementing / configuring NatQuery. This is due to the fact that while NatWorks does deliver example JCL / Script templates with a NatQuery installation, these examples will not typically execute as provided. This is due to the simple reason that the requirements for JCL / Scripts can vary radically between customer sites, with things like how Natural is invoked, file naming standards, etcetera. Due to the myriad of differences that may exist between a NatWorks-provided example and a given site's requirements, the process of converting an example template into a fully functioning JCL / Script template may require several iterations in order to develop a fully functioning JCL / Script template.

Fortunately, once the first JCL / Script template is made operational, the remaining script templates that may be needed to be built become vastly easier to develop, as the lesson(s) learned in what is needed to be changed in the first template can be applied to the development of subsequent JCL / Script templates.

To support batch processing, a NatQuery installation will by default support several different types of "standard" JCL / Script templates. Additionally, NatQuery provides the ability to allow an Administrator to create their own "custom" templates, and then allows for these customer templates to be associated with given NatQuery functions.

This section covers the following aspects of JCL / Scripts:

- Handling Standard JCL / Script Templates,
- Platform Considerations for JCL / Script Handling,
- Description(s) of Standard JCL / Script Templates,
- JCL / Script Execution,
- Creating Custom JCL / Script Templates, and
- Handling JCL / Script Using External Editors

Handling Standard JCL / Script Templates

The creation and manipulation of JCL / Script templates primarily occurs through the use of the Administer JCL / Script function. Alternatively, JCL / Script templates can be edited using tools such as Notepad as described in the section Handling JCL / Script Using External Editors.

To perform JCL / Script maintenance, the Administrator would perform the following steps:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Instructions on how to start NatQuery can be found in the section entitled Starting NatQuery.

2. **Invoke the Administer FTP JCL / Script function**
   With no queries open on the NatQuery desktop, click on the **Administer** drop-down menu, then click on **Environment Configuration / Server Connection Configuration / JCL** / **Script Information**. This will invoke the **Administer JCL / Script** function.

3. **Select JCL / Script Template that Requires Handling**
   By default, the **Administer JCL / Script** window will display the **Production Request Process** template.

   To select an alternate JCL / Script template, the administrator will use the combo box located in the upper-left corner entitled **Server JCL / Script Component**.

   If the JCL / Script template needing to be handled does not currently exist in the currently designated Environment Configuration path, then when the JCL / Script is selected, the text box in the center of the window will display a message similar to "This template does not contain any JCL/Script information". In this case, the administrator can click the button labeled **Copy from Example Template** to copy the installation-provided example JCL / Script template for the selected template into the current Environment Configuration path and therefore into the text box in the center of the window.

4. **Make Appropriate Change(s) to JCL / Script template**
   Through the use of the **Administer JCL / Script** function, the Administrator would make the appropriate changes, and after all changes were completed, the template would be saved by clicking on the **Save** button.

   Specific Help on using the **Administer JCL / Script** function is available by clicking on the **Help** button on this window.

   If after clicking the Save button the administrator wishes to modify other JCL / Script templates, they can be individually handled now by selecting the appropriate JCL / Script template using the **Server JCL / Script Component** combo box.

   If a change needed to the JCL / Script template involves the use of a **Dynamic Substitution Variable**, these variables are listed in the combo box labeled **Available**

Dynamic Substitution Variables. The easiest way to handle utilizing a **Dynamic Substitution Variable** is to first locate it in this list by clicking on the desired variable; this action will highlight the selected variable. Once highlighted, the user will hold down the "CTRL" key and the "C" key, and then release both – these actions will copy the highlighted variable onto the Windows clipboard. The administrator may then position the cursor within the central text box and click on the location where the Dynamic Substitution Variable is needed. The user can then hold down the "CTRL" key and the "V" key, and then release both – these actions will copy the selected Dynamic Substitution variable from the Windows clipboard into the text box at the point specified by the cursor.

5. **Close Administer JCL / Script Window**
   When all required JCL / Script changes are complete, the Administrator would close the **Administer JCL / Script** window by clicking the **OK** button.

   As this window closes, NatQuery will typically produce a message box similar to the one seen in Figure 3 (**Verification Question**). It is suggested that the Administrator should respond to this prompt by clicking the :Yes" button.

6. **Review Verify Environment Configuration Report**
   Once the Verify Environment Configuration report has been executed, this function will produce a report concerning the status of the current Environment Configuration.

   When NatQuery "verifies" JCL / Script, NatQuery can actually do little more than check for the existence of a given Production template, as it is beyond the scope of NatQuery to attempt to edit the "correctness" of any given template.

Platform Considerations for JCL / Script Handling

While the basic processing of any given JCL or Script template's functioning is more or less identical across all desired target platforms, there are differences. This section attempts to highlight these differences in the following sections:

- Introduction of a Generated Program Into Natural
- Success / Error Reporting (Mainframe Servers)
- Success / Error Reporting (Open System Servers)
- Output File Handling (Mainframe Server)
- Log File Handling (Open System Servers)
- Request represented by Multiple Server Files (Open System Servers)
- Handling Job Steps (Open System Servers)

Introduction of a Generated Natural Program into Natural
Against mainframes systems, NatQuery currently supports the introduction of a Natural program into Natural on the mainframe through the ability to invoke the Natural editor in batch (I.E. using the “E” line command).

Against non-mainframe systems, NatQuery supports the introduction of a Natural program through the use of a NatQuery installation provided program called NQYPNT05, which in turn utilizes the Natural sub-program USR0080N.

Success / Error Reporting (Mainframe Servers)

In order to provide feedback on the execution status of a request to a submitting user, NatQuery uses a simple logging mechanism. On the Natural server platform each user is assigned a unique NatQuery Log File (Remote Log File), and this Remote Log File must be updated with the execution status of each submitted request. This updating will provide the information as to whether the request completed successfully (a status of “**DONE**”), or the request failed (a status of “**FAILED**”).

A “**DONE**” message (I.E. the successful completion of a request) will be handled directly by a generated data extraction program, or if no generated program is being run (for example a SYSTRANS / SYSOBJH request) then the log file is updated by an installation provided Natural program called NQYP0003. In such a case, NQYP0003 will be executed as a job step that is conditionally executed if the Natural batch job step executes successfully (I.E. Condition Code / Return Code interrogation).

A “**FAILED**” message (I.E. a request not executing successfully) is always handled by the execution of a program named NQYP0002. In such case, NQYP0002 will be executed as a job step that is conditionally executed if the Natural batch job step executes successfully (I.E. Condition Code / Return Code interrogation).

Success / Error Reporting (Open System Servers)

In order to provide feedback on the execution status of a request to a submitting user, NatQuery uses a simple logging mechanism. On the Natural server platform each user is assigned a unique NatQuery Log File (remote Log File), and this Remote Log File must be updated with the execution status of each submitted request. This updating will provide the information as to whether the request completed successfully (a status of "**DONE**"), or the request failed (a status of "**FAILED**").

A "**DONE**" message (I.E. the successful completion of a request) will be handled directly by a generated data extraction program, or if no generated program is being run (for example a SYSTRANS / SYSOBJH request) then the log file is updated by an installation / configuration-provided Natural program called NQYPNT03. In such a case, NQYPNT03 will be executed as a command passed via CMSYNIN: With the use of the Natparm CC=ON, then NQYPNT03 would only be executed if the SYSTRANS / SYSOBJH completed without error.

A "**FAILED**" message (I.E. a request not executing successfully) is always handled by the execution of a program named NQYPNT06. In such case, the use of the Natparm ETA=NQYPNT06 will insure that NQYPNT06 will be executed in the event the requested process did not complete successfully.

Output File Handling (Mainframe Servers)

In mainframe environments, if a file already exists and this file is written to, then the file will typically be appended to (depending on the use of the DISP parameter). To accommodate the "re-use" of file names that will contain the output of a user's extract request, and to also provide a single generic JCL template for a given operation, it is suggested that any NatQuery JCL template perform as a first step a utility that will delete (and then possibly re-allocate) a new output dataset each time the given JCL template is executed. In most cases this will be the use of IDCAMS, IEFBR14 or a similar utility.

For non-mainframe systems such as UNIX or Windows, this is not a consideration as write operations will typically over-write an existing file.

Log File Handling (Open System Servers)

In non-mainframe environments, if a file already exists and this file is then written to, then the file will typically be over-written. To provide for the ability to have the existing contents of a user's Remote Log File be preserved (I.E. a user may have multiple active requests) from one request to the next, NatQuery handles this by typically writing the results of a given execution to a "temporary" Log File. This temporary Log entry will then typically be appended to the user's Remote Log File using a copy command; this command is typically issued by the generated Natural via USR1052N with the physical command passed being administratively configurable.

For UNIX systems, this could be accomplished using the command:

cat ***perm-log-file*** >> ***temp-log-file***

For Windows systems, this could be accomplished using a command similar to:

cmd.exe /c copy /b ***perm-log-file*** +/b ***perm-log-file*** /b ***temp-log-file*** /b

To allow for maximum flexibility, the generated Natural will refer to ***perm-log-file*** and the ***temp-log-file*** through the use of environment variables CMWRK20 and CMWRK21 respectively. It is therefore usual to have these environment variables be set in a NatQuery batch or shell script template.

Request Represented by Multiple Server Files (Open System Servers)

In mainframe environments, a NatQuery processing request is typically a single file with this file being the JCL. This JCL will therefore contain the appropriate execution statements and references, and will additionally contain the NatQuery-generated Natural program(s) that will often be central to processing a given request.

This contrasts against non-mainframe environments where Software AG typically recommends the use of a central Batch or Script file, as well as additional and separate files for CMOBJIN (used to supply data statements) and CMSYNIN (used to supply both data statements and commands), with yet another file containing any NatQuery-generated Natural program that will assist in the processing of the request.

When dealing with mainframe JCL then, the administrator will be using the **Administer JCL / Script** function to administer a file that represents the single JCL stream that will be executed, with any NatQuery-generated Natural program being introduced into Natural on the server through an invocation of the Natural editor in batch (I.E. using the "E" command subsequently followed by the ".E" command).

When dealing with non-mainframe Script however, the administrator will be using the **Administer JCL / Script** function to administer a single file that NatQuery will then subsequently "split" into at least 3 and in most cases 4 separate files that will then typically be introduced onto the server. NatQuery accomplishes this "split" by examining the single Script file for "tags" that delineate each separate component (the batch /script file, CMOBJIN, CMSYNIN and the Natural Program file). These "split" tags are as follows:

- ***!*CMSYNIN INPUT FOLLOWS THIS LINE**
  This tag is usually the first line of a script file, and indicates that the lines following this (the tag itself is dropped), up until the next line that is either a "*!**command*" or "*!*!*!*!*" statement will become the CMSYNIN file on the server.

- ***!*CMOBJIN INPUT FOLLOWS THIS LINE**
  This tag indicates that the lines following this line (the tag itself is dropped), up until the next line that is either a "*!**command*" or "*!*!*!*!*" statement will become the CMOBJIN file.

- ***!*SCRIPT FOLLOWS THIS LINE**
  This tag indicates that the lines following this line (the tag itself is dropped), up until the next line that is either a "*!**command*" or "*!*!*!*!*" statement will become the Batch (.bat) or Script (.sh) file.

- ***!*!*!*!***
  This tag indicates that the lines following this line (the tag itself is dropped), up until the end of file, will typically contain the NatQuery-generated Natural program.

Handling Job Steps (Open System Servers)

When dealing with JCL for mainframes or Script for Unix systems, the necessary individual job steps are easily handled in JCL / Script construction. In Windows (I.E. when the Natural Server resides on a Windows platform) environments it is a bit more nebulous, as each "step" in a batch file is essentially a command line (with or without optional parameters), with individual steps being delineated by End-Of-Block (EOB) or Carriage Return (CR).

To allow for job steps within a Windows batch file, NatQuery supports the concept of steps through the use of the "***!*STEP**" tag.

The "***!*STEP**" tag may be used anywhere within the SCRIPT section of a template, with the ability to have as many "***!*STEP**" statements as needed to achieve multi-step jobs with a Windows server environment.

Description(s) of Standard JCL / Script Templates

These standard JCL / Script templates that NatQuery supports are:

- Production Sequential Process,
- Production ADABAS Process,
- Production Predict Process,
- Production E-Mail Process,
- Production ADACMP Process (Decompress Infile),
- Production ADAULD Process (Unload from ADABAS File),
- Production DWH Process,
- Production ADAULD Process (Unload from ADABAS Backup),
- Production Summary Process,
- Production DB2 Process,
- Production NatCDC Process,
- Production Request Process,
- Production Special Process (SYSTRANS / SYSOBJH), and
- Production XML Server Process

In most cases, only a few of the above named templates will be needed to support any given installation.

The unique processing requirements of each of these templates are described in following sections.

Production Sequential Process

The **Production Sequential Process** template is used in any situation where NatQuery is used to create an extract when one of the files selected for extraction is a Sequential File. Typically this template is automatically selected for use by NatQuery at the point that it detects the user's query references is a sequential file.

The **Production Sequential Process** template has the following attributes:

Processing Overview:

Execution of a NatQuery-generated Natural program, with the generated program substituted into &&USER-EXTRACT-PROGRAM. The generated program will expect to have its primary read be resolved from a sequential input file, with this sequential file having a fixed-length record layout as described in a NatQuery DDM.

Handling a sequential file requires that NatQuery have or be given a DDM that matches the layout of the Sequential File. In some cases a sequential DDM can be generated by NatQuery as a result of creating a NatQuery sequential extract, in other cases a Sequential DDM can be built using a Build / Edit DDM function of NatQuery, and in other situations a DDM representing a sequential file can be built elsewhere and then Imported into NatQuery.

Internal File Name:

server_typeA.PRD	(Production Template)
server_typeA.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1	Permanent User Log File
Work File 2	Extracted Data Output File
Work File 3	Sequential Input File

Work Files Used (non-Mainframes):

Work File 1	Temporary Log File
Work File 2	Extracted Data Output File
Work File 3	Sequential Input File
Work File 20	Temporary Log File
Work File 21	Permanent User Log File

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Introduce and run generated Natural program
Step 3	Execute NQYP0002 if Step 2 fails (handle user's log)

Job Steps (UNIX):

Step 1	Introduce and run generated Natural program
Step 2	Execute program “unix2dos” against CMWRK20
Step 3	Execute program “unix2dos” against CMWRK21

Steps 2 and 3 are only required when the Communication Mode is PC Network

Job Steps (Windows)

Step 1	Set CMWRK20 (Environ. Var.) to Temporary Log File
Step 2	Set CMWRK21 (Environ. Var.) to Permanent Log File
Step 3	Introduce and run generated Natural program

Production ADABAS Process

The **Production ADABAS Process** template is used to read a Sequential input file and then process some or all of the data in that Sequential File into ADABAS. Transactions against ADABAS can be optionally processed in a number of ways, as designated through the **Process Data Into ADABAS** function.

This JCL / Script is used solely by the **Process Data Into ADABAS** function, which is an Administrator-only function.

The **Production ADABAS Process** template has the following attributes:

Processing Overview:

Execution of a NatQuery-generated Natural program, with the generated program substituted into &&USER-EXTRACT-PROGRAM. The generated program will expect to have its primary read be resolved from a sequential input file, with this file having a fixed-length record layout as described in a NatQuery DDM.

Handling a sequential file requires that NatQuery have available a DDM that matches the layout of the sequential File. In some cases a sequential DDM can be generated by NatQuery as a result of creating a NatQuery extract, in other cases a sequential DDM can be built using a **Build / Edit DDM** function of NatQuery, and in other situations a DDM representing a sequential file can be built elsewhere and then Imported into NatQuery.

The **Production ADABAS Process** is not intended to write any entry to any log file, either the Remote Log File or the Local Log File.

Internal File Name:

server_type**B.PRD**	(Production Template)
server_type**B.EXM**	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 3	Sequential Input File

Work Files Used (non-Mainframes):

Work File 3	Sequential Input File

Job Steps (Mainframes):

Step 1	Introduce and run generated Natural program

Job Steps (UNIX):

Step 1	Introduce and run generated Natural program

Job Steps (Windows)

Step 1 Introduce and run generated Natural program

Production Predict Process

The **Production Predict Process** template is used to extract all field-level information that may exist in Predict for a given file, for the purpose of utilizing this information as text within a NatQuery Environment Configuration.

The **Production Predict Process** template is automatically utilized by NatQuery at any time that an administrator uses the **Predict** / **Request Predict Information** function.

The **Production Predict Process** template has the following attributes:

Processing Overview:

The **Production Request Process** template will logon on to the SYSDIC library where it will invoke MENU so as to pass generated commands as passed via &&PREDICT-COMMANDS.

The output of the Predict function expects to output the information to a report (I.E. CMPRT01) with CMPRT01 being directed at a disk file instead of a printer.

Internal File Name:

server_typeD.PRD	(Production Template)
server_typeD.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

CMPRT01	Predict Filed-Level Information
Work File 1	Permanent User Log File

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Logon SYSDIC, execute MENU, and pass parameters
Step 3	Execute NQYP0004 if Step 2 is OK (handle user’s log)
Step 3	Execute NQYP0002 if Step 2 fails (handle user’s log)

Production E-Mail Process

The **Production E-Mail Process** template is used in the situation where an e-mail client can be invoked in the server environment within a JCL or script stream, such that the results of a user's e-mail request (which will exist on the server as a fixed-length file) can be automatically e-mailed.

This JCL / Script template is only available for use if a user chooses an **Extract Type** of **Download to PC File**, the administrator has first turned on NatQuery's E-Mail handling as well as set up an E-Mail template, and the user selects **Production E-Mail Process** from the **Output Handling Options** combo box.

Further information on the E-Mail handling of NatQuery extracted data can be found in the section entitled Enabling E-Mail handling of Extracted Output.

The **Production E-Mail Process** has these attributes:

Processing Overview:

The processing of the Production E-Mail Process JCL / Script template will be identical to the Production Request Process template, with the addition of an extra step the takes the output file and invokes an SMTP or similar E-Mail client utility.

Internal File Name:

server_typeE.PRD	(Production Template)
server_typeE.EXM	(Example Template – not always provided)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1 (Step 2)	Permanent User's Log File (&&USER-LOG-FILE)
Work File 2 (Step 2)	Extracted Data Output File (&&USER-OUTPUT-FILE)
Work File 1 (Step 4)	Permanent User's Log File (&&USER-LOG-FILE)

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute NatQuery-generated Natural Program
Step 3	Execute E-Mail software if Step 2 Succeeds
Step 4	Execute NQYP0002 if Step 2 Fails (Handle User's Log)

Production ADACMP Process (Decompress Infile)

The **Production ADACMP Process** template is used in the situation where an Administrator wishes to extract data from a single ADABAS file using the ADACMP utility of ADABAS, with the output of the execution of this utility then being further processed by a generated Natural program.

For Mainframe systems, the generated ADACMP process is intended to only support the execution of the ADACMP utility with the *Decompress Infile* option (please refer to the ADABAS Utilities Manual for further information).

The **Production ADACMP Process** template has the following attributes:

Processing Overview:

Execution of the ADABAS Utility ADACMP, with the ADACMP parameters being generated into &&ADACMP-PARMS. The output of ADACMP is then further processed by a NatQuery-generated Natural program so as to perform any data conversions or drop any un-needed fields.

Log File handling is not expected with the Production ADACMP Process.

Internal File Name:

server_typeF.PRD	(Production Template)
server_typeF.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

DDAUSBA (step 2)	ADACMP Output File
DDFEHL (step2)	ADACMP Rejects File
Work File 1 (step 3)	ADACMP Output File
Work File 2 (step 3)	Final Data Output File (&&ADACMP-FINAL)

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute ADACMP using generated parameters
Step 3	Execute generated Natural program

Production ADAULD Process (Unload from ADABAS File)

The **Production ADAULD Process** template is used in the situation where an Administrator wishes to extract data from a single ADABAS file directly from ADABAS using the ADAULD utility, with the output of this utility then being further processed by a NatQuery-generated Natural program.

The **Production ADAULD Process** template is used automatically by NatQuery at any time an administrator creates a single-file extract request and then sends this extract request to the server using an **Extract Type** of **ADAULD Extract**, and once in the **ADAULD (Unload) ADABAS File** function – the administrator selects the **Unload from ADABAS** option.

There are two NatQuery processes that support the use of ADAULD; the process described here that details extraction via ADAULD directly from ADABAS, and the other supports extraction via ADAULD directly from an ADABAS Backup tape produced by ADASAV. For information concerning NatQuery's ability to support ADAULD from an ADABAS Backup tape, please refer to the section Production ADAULD Process (Unload from ADABAS Backup).

The **Production ADAULD Process** template has the following attributes:

Processing Overview:

Execution of the ADABAS Utility ADAULD, with the ADAULD parameters being generated into &&ADAULD-PARMS and passed via DDKARTE. The output of ADAULD is then further processed by a NatQuery-generated Natural program so as to perform any data conversions or drop any un-needed fields.

In current versions of NatQuery - Log File handling is not handled – so any such extraction will require manual monitoring.

Internal File Name:

server_typeG.PRD	(Production Template)
server_typeG.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

DDOUT1 (Step 2)	ADAULD Output File (&&ADAULD-TEMP)
EBAND (Step 3)	ADAULD Output File (&&ADAULD-TEMP)
AUSBA (Step 3)	ADACMP Output File (&&ADACMP-OUTPUT)
Work File 1 (Step 4)	ADAULD Output File (&&ADACMP-OUTPUT)
Work File 2 (Step 4)	Final Data Output File (&&ADACMP-FINAL)

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute ADAULD using generated parameters
Step 3	Execute ADACMP using generated parameters
Step 4	Execute generated Natural program

Production DWH Process

The **Production DWH Process** is used to support Data Warehousing initiatives, usually in combination with the use of ETL tools such as IBM's Data Stage Suite (formally Ascential Software's Web Sphere Data Stage Suite), Pervasive Software (formally Data Junction), Informix and others. A primary differentiator between a **Production DWH Process** and other data extraction processing is that using the **Production DWH Process** allows for the automatic generation of a DDM that describes the layout of the extract data file – with this processing also allowing for the generation of "interface files" such as Cobol File Definitions (CFDs) or Data Stage Exchange Files (DSXs) that allow ETL tools to understand a sequential file's data layout.

While primarily designed to support integration to ETL tools, the **Production DWH Process**, through its ability to automatically generate DDMs, allows for what could be considered a "Poor Man's Data Warehouse". For example, NatQuery could be used to create a sequential file on the server platform, which can then be given to users as a data source, with NatQuery being able to generate Natural that will process this sequential file.

The **Production DWH Process** template is automatically selected for use when a user selects an **Extract Type** of **DWH Software Extract**.

The **Production DWH Process** template has the following attributes:

Processing Overview:

Execution of a NatQuery-generated data extraction program, with the optional capability to generate a DDM that maps the sequential extract data, as well as optionally generated CFD or DSX interface files.

When using an **Extract Type** of **DWH Software Extract**, NatQuery Log File handling is optional.

Internal File Name:

server_typeH.PRD	(Production Template)
server_typeH.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1 (Step 2)	Permanent User's Log File (&&USER-LOG-FILE)
Work File 2 (Step 2)	Extracted Data Output File (&&USER-OUTPUT-FILE)
Work File 3 (Step 2)	Totals File (opt - &&USER-OUTPUT-FILE.TOTALS)
Work File 1 (Step 3)	Permanent User's Log File (&&USER-LOG-FILE)

Work Files Used (non-Mainframes):

Work File 1	Temporary User's Log File
Work File 2	Extracted Data Output File (&&USER-OUTPUT-FILE)
Work File 3	Totals File (opt - &&USER-OUTPUT-FILE.TOTALS)
Work File 20	Temporary User's Log File

Work File 21	Permanent User’s Log File (&&USER-LOG-FILE)

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute NatQuery-generated Natural Program
Step 3	Execute NQYP0002 if Step 2 Fails (Handle User’s Log)

Job Steps (UNIX):

Step 1	Execute NatQuery-generated Natural Program
Step 2	Execute “unix2dos” on CMWRK21 (PC Network only)
Step 3	Execute “unix2dos” on CMWRK2 (PC Network only)
Step 4	Execute “unix2dos” on CMWRK3 (PC Network only)

Job Steps (Windows):

Step 1	Set CMWRK20 (Environ. Var.) to Temporary Log File
Step 2	Set CMWRK21 (Environ. Var.) to Permanent Log File
Step 3	Introduce and run generated Natural program

Production ADAULD Process (Unload from ADABAS Backup)

The **Production ADAULD Process** template is used in the situation where an Administrator wishes to extract data from a single ADABAS file from an ADABAS Backup tape using the ADAULD utility, with the output of this utility then being further processed by a NatQuery-generated Natural program.

The **Production ADACMP Process** template is used automatically by NatQuery at any time an administrator create a single-file extract request and then sends this extract request to the server using an **Extract Type** of **ADAULD Extract**, and once in the **ADAULD (Unload) ADABAS File** function – the administrator selects the **Unload from ADASAV** option.

There are two NatQuery processes that support the use of ADAULD; the process described here that details extraction via ADAULD from an ADABAS Backup, and the other supports extraction via ADAULD directly from an ADABAS. For information concerning NatQuery's ability to support ADAULD directly from an ADABAS file, please refer to the preceding section entitled Production ADAULD Process (Unload from ADABAS File).

The **Production ADAULD Process** template has the following attributes:

Processing Overview:

Execution of the ADABAS Utility ADAULD, with the ADAULD parameters being generated into &&ADAULD-PARMS. The output of ADACMP is then further processed by a NatQuery-generated Natural program so as to perform any data conversions or drop any un-needed fields.

In current versions of NatQuery - Log File handling is not handled – so any such extraction will require manual monitoring.

Internal File Name:

server_typeI.PRD	(Production Template)
server_typeI.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

DDSAVE (step 2)	ADASAV File (&&ADAULD-SAVETAPE)
DDPLOG (step2)	PLOG File (&&ADAULD-PLOG; Online Save only)
DDOUT1 (step 2)	ADAULD Output File (&&ADAULD-TEMP)
EBAND (step 3)	ADAULD Output File (&&ADAULD-TEMP)
AUSBA (step 3)	ADACMP Output File (temp file)
Work File 1 (step 4)	ADACMP Output File (temp file)
Work File 2 (step 4)	Final Data Output File (&&ADACMP-FINAL)

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute ADAULD using generated parameters

Step 3	Execute ADACMP using generated parameters
Step 4	Execute generated Natural program

Production Summary Process

The **Production Summary Process** is used as a secondary process that works against a sequential file created by a primary NatQuery-generated extract process. This sequential file is first sorted, and then this file is processed by a NatQuery-generated Natural program that will read the sorted file and produce summarized output according to user-specified requirements.

The **Production Summary Process** is used automatically by NatQuery to produce summarized results at the point in time that a user has submitted a "detail data" extract request, this request executes such that the **Check Server** function shows that the detail extract has been completed successfully.

To review, the **Production Summary Process** is used in this fashion:

1. A User submits a data extraction request, and when the request is sent to the server, the user selects **Summary Processing** on the **Send To Server** window. Selection of Summary Processing is done independently of setting the **Extract Type**.

   Internally, NatQuery responds to this by sending the user's initial request to the server using the **Production Request Process** template. In addition to generating the program needed to produce the "detail" data, NatQuery additionally generates a second Natural program that is designed to produce summarized results on the sorted output of the detail data request.

2. When the detail request is "**DONE**" as seen in the **Check Server** function, the user may download the detail data or they may optionally skip the download of the detail data. Either way, NatQuery will prompt the user as to whether they wish to submit the **Summary Process** or not.

   If the user requests **Summary Process**, then NatQuery will pick up the NatQuery-generated Natural Summary program that was previously generated and will then use this program in conjunction with the **Production Summary Process** template to create a new extract request.

   Since the **Summary Process** is based on utilizing the detail extract data file created by a previous and separate request, NatQuery will submit the **Summary Process** into the same "request slot" just freed by the detail request. In this way, the detail extract file is preserved.

The **Production Summary Process** has the following attributes.

Processing Overview:

A multi-step JCL / Script that will first execute a Sort of a fixed-length sequential file (the "detail data) followed by the execution of a NatQuery-generated Natural program that will produce summarized results from the detail data.

Internal File Name:

server_typeJ.PRD (Production Template)

server_typeJ.EXM (Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

SORTIN (step 2)	User Output File (&&USER-OUTPUT-FILE)
SORTOUT (step 2)	Sorted Output File (&&USER-OUTPUT-FILE.SORTED)
Work File 1 (step 4)	User Permanent Log File (&&USER-LOG-FILE)
Work File 2 (step 4)	User Output File (&&USER-OUTPUT-FILE)
Work File 3 (step 4)	Sorted Output File (&&USER-OUTPUT-FILE.SORTED)
Work File 1 (step 5)	User Permanent Log File (&&USER-LOG-FILE)

Work Files Used (non-Mainframes):

Work File 1	Temporary User's Log File
Work File 2	User-Output-File (&&USER-OUTPUT-FILE)
Work File 3	Sorted Output File (&&USER-OUTPUT-FILE-SORT)
Work File 20	Temporary User's Log File
Work File 21	Permanent User's Log File (&&USER-LOG-FILE)

Job Steps (Mainframes):

Step 1	IDCAMS / IEFBR14	
Step 2	Execute System SORT	
	In	&&USER-OUTPUT-FILE
	Out	&&USER-OUTPUT-FILE.SORTED
Step 3	IDCAMS / IEFBR14	
	delete	&&USER-OUTPUT-FILE
Step 4	Execute generated Natural program	
Step 5	Execute NQYP0002 if Step 4 Fails (Handle User's Log)	

Job Steps (UNIX):

Step 1	Execute System SORT	
	In	&&USER-OUTPUT-FILE
	Out	&&USER-OUTPUT-FILE-SORT
Step 2	RM (Delete) &&USER-OUTPUT-FILE	
Step 3	Execute generated Natural program	
Step 4	Execute "unix2dos" on CMWRK21 (PC Network only)	
Step 5	Execute "unix2dos" on CMWRK02 (PC Network only)	

Job Steps (Windows):

Step 1	Set CMWRK20 (Environ. Var.) to Temporary Log File	
Step 2	Set CMWRK21 (Environ. Var.) to Permanent Log File	
Step 3	Execute System SORT	
	In	&&USER-OUTPUT-FILE
	Out	&&USER-REQUEST-FILE-SORT
Step 4	Execute generated Natural program	

Production DB2 Process

The **Production DB2 Process** template is used in the situation where an administrator would like to extract data from ADABAS and immediately load this data into DB2 using the DB2 Loader utility invoked in batch.

This JCL / Script template is used exclusively when a user is an Administrator, and an **Extract Type** of **DB2 Loader** is utilized.

The **Production DB2 Process** has these attributes:

Processing Overview:

The processing of the Production DB2 Loader JCL / Script template will be identical to the Production Request Process template, with the addition of an extra step the takes the output file and invokes the DB2 Loader.

Internal File Name:

server_typeL.PRD	(Production Template)
server_typeL.EXM	(Example Template – not always provided)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1 (Step 2)	Permanent User’s Log File (&&USER-LOG-FILE)
Work File 2 (Step 2)	Extracted Data Output File (&&USER-OUTPUT-FILE)
Work File 1 (Step 4)	Permanent User’s Log File (&&USER-LOG-FILE)

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute NatQuery-generated Natural Program
Step 3	Execute DB2 Loader utility if Step 2 Succeeds
Step 4	Execute NQYP0002 if Step 2 Fails (Handle User’s Log)

Production NatCDC Process

The **Production NatCDC Process** template is used to process a NatCDC PLOG Process. A **Production NatCDC Process** JCL / Script template is only used / required when a license for NatCDC is purchased or NatCDC is otherwise available.

The **Production NatCDC Process** has the following attributes.

Processing Overview:

A multi-step JCL / Script that will expect to execute the NatQuery supplied NatCDC utility, a system SORT, and then NatQuery-generated processing program(s).

NatCDC Processing does not expect to handle any log file and this processing requires that output be made available by one of several ADABAS PLOG utilities for input.

Internal File Name:

server_typeP.PRD	(Production Template)
server_typeP.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1 (Step 2)	NatCDC Input parameters (&&CDC-PARMS)
Work File 2 (Step 2)	ADASEL / ADACDC output (&&CDC-ADASEL)
Work File 3 (Step 2)	Interim file 1 (&&CDC-I-DSNAME1)
Work File 1 (Step 4)	SORTed Output File (&&CDC-I-DSNAME2)
Work File 2 (Step 4)	Final Data Output (&&CDC-F-DSNAME)
Work File 3 (Step 4)	Optional Totals File (&&CDC-TOTALS)

Work Files Used (non-Mainframes):

Work File 1 (Step 1)	NATCDC Input parameters (&&CDC-PARMS)
Work File 2 (Step 1)	NATPLP Output (NATCDC Input)
Work File 3 (Step 1)	Interim file 1 (NATCDC Output) (&&CDC-I-DSNAME1)
Work File 4 (Step 1)	ADAPLP output (NATPLP Input) (&&CDC-ADASEL)
Work File 1 (Step 2)	NATSORT / System SORT Input (&&CDC-I-DSNAME1)
Work File 2 (Step 2)	NATSORT / System SORT Output (&&CDC-I-DSNAME2)
Work File 1 (Step 3)	NATSORT / System SORT Output (&&CDC-I-DSNAME2)
Work File 2 (Step 3)	Final Data Output (&&CDC-F-DSNAME)
Work File 3 (Step 3)	Optional Totals File (&&CDC-TOTALS)

Job Steps (Mainframes):

Step 1	IDCAMS / IEFBR14 (optional)
Step 2	Execute NatCDC using generated parameters
Step 3	Execute System SORT on NatCDC output
Step 4	Execute generated Natural against SORT output
Step 5	IDCAMS / IEFBR14 (optional)

Job Steps (UNIX):

Step 1	Execute NatPLP and NatCDC using generated parms
Step 2	Execute NATSORT / System SORT
Step 3	Execute generated Natural program(s)
Step 4	Execute unix2dos against CMWRK02 (PC Network Only)
Step 5	Execute unix2dos against CMWRK02 (PC Network Only)

Job Steps (Windows):

Step 1	Execute NatPLP and NatCDC using generated parms
Step 2	Execute NATSORT / System SORT
Step 3	Execute generated Natural program(s)

Production Request Process

The **Production Request Process** template is the main JCL / Script template of NatQuery and is used for several different **Extract Types**, such as:

- Download to **PC File** (fixed-length, delimited or non-delimited)
- Download into **Excel**
- **SourcePoint** Extract
- Download into **Access**
- RDBMS Loading (such as **Oracle** or **SQL Server**)
- Download to **XML** File
- Data Discovery / Analysis functions

The **Production Request Process** template has the following general attributes.

Processing Overview:

Execution of a NatQuery-generated data extraction program to create a fixed-length output file, with a user's log file being updated with the process results.

Internal File Name:

server_typeR.PRD	(Production Template)
server_typeR.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1 (Step 2)	Permanent User's Log File (&&USER-LOG-FILE)
Work File 2 (Step 2)	Extracted Data Output File (&&USER-OUTPUT-FILE)
Work File 1 (Step 3)	Permanent User's Log File (&&USER-LOG-FILE)

Work Files Used (non-Mainframes):

Work File 1	Temporary User's Log File
Work File 2	Extracted Data Output File (&&USER-OUTPUT-FILE)
Work File 20	Temporary User's Log File
Work File 21	Permanent User's Log File (&&USER-LOG-FILE)

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute NatQuery-generated Natural Program
Step 3	Execute NQYP0002 if Step 2 Fails (Handle User's Log)

Job Steps (UNIX):

Step 1	Execute NatQuery-generated Natural Program
Step 2	Execute "unix2dos" on CMWRK21 (PC Network only)
Step 3	Execute "unix2dos" on CMWRK2 (PC Network only)

Job Steps (Windows):

Step 1	Set CMWRK20 (Environ. Var.) to Temporary Log File
Step 2	Set CMWRK21 (Environ. Var.) to Permanent Log File
Step 3	Introduce and run generated Natural program

Production Special Process (SYSTRANS / SYSOBJH)

The **Production Special Process** template is used to invoke the execution of a SYSTRANS or SYSOBJH utility for the purpose of downloading a DDM or FDT.

This JCL / Script is used solely by the **Download DDM** and **Download FDT** functions.

The **Production Special Process** template has the following attributes:

Processing Overview:

Execution of a NatQuery-generated Natural program, with the generated program substituted into &&USER-EXTRACT-PROGRAM. The generated program will expect to have its primary read be resolved from a sequential input file, with this file having fixed-length record layout as described in a NatQuery DDM.

Internal File Name:

server_typeU.PRD	(Production Template)
server_typeU.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1 (Step 2)	SYSTRANS Output File
Work File 3	SYSTRANS Temporary File
Work File 1 (Step 3)	Permanent User Log File
Work File 1 (Step 4)	Permanent User Log File

Work Files Used (non-Mainframes):

Work File 1	SYSOBJH Output File
Work File 2-15	SYSOBJH Temporary File(s)
Work File 20	Temporary Log File
Work File 21	Permanent User Log File

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute SYSTRANS using generated parameters
Step 3	Execute NQYP0003 if Step 2 succeeds (handle user's log)
Step 4	Execute NQYP0002 if Step 3 fails (handle user's log)

Job "Steps" (UNIX):

"Step" 1	Execute SYSOBJH using generated parameters
"Step" 2	execute program "unix2dos" against CMWRK20
"Step" 3	execute program "unix2dos" against CMWRK21

Steps 2 and 3 are only required when the Communication Mode is PC Network

Job "Steps" (Windows)

“Step” 1	Set CMWRK20 (Environ. Var.) to Temporary Log File
“Step” 2	Set CMWRK21 (Environ. Var.) to Permanent Log File
“Step” 3	Execute SYSOBJH using generated parameters

Production XML Server Process

The **Production XML Server Process** is used to produce an ADABAS extract in XML format directly on the server as a result of the processing of a single JCL / Script. This template is used exclusively with the **Extract Type** of **Extract to XML on Server**.

The **Production XML Server Process** has the following attributes:

Processing Overview:

A multi-step JCL / Script that will first execute a typical NatQuery data extract, however the Production XML Server Process will additionally generate a second Natural program that is designed to read the contents of the fixed-length output file and then convert this into XML.

With the Production XML Server Process there is no provision to have NatQuery generate any XSL for the XML file. To have NatQuery produce both an XML file and corresponding XSL file, the **Extract Type** of **Download To XML File** should be used.

Internal File Name:

server_typeX.PRD	(Production Template)
server_typeX.EXM	(Example Template)

server_type can be MVS, VSE, UNIX or NT

Work Files Used (Mainframes):

Work File 1 (Step 2)	User Permanent Log File
Work File 2 (Step 2)	Interim Output File
Work File 3 (Step 2)	XML Output File
Work File 1 (Step 3)	User Permanent Log File

Job Steps (Mainframes):

Step 1	IDCAMS or IEFBR14 (optional)
Step 2	Execute generate Natural Program(s)
Step 3	Execute NQYP0002 if Step 2 fails (handle user’s log)

JCL / Script Execution

The execution of a NatQuery-generated JCL / Script is the event that actually produces the requested results, which is usually extracted data.

The manner in which these JCL / Scripts get physically executed depends on the platform upon which the Natural server resides, as well as the selected method of execution.

Please refer to the following section appropriate to your site:

- Execution of JCL on Mainframes, or
- Execution of Scripts on non-Mainframes

Execution of JCL on Mainframes

When integrating against mainframes, NatQuery generates JCL to accomplish a given process. NatQuery will support moving this JCL into a mainframe environment using File Transfer Protocol (FTP).

In most cases, it is anticipated that NatQuery will be configured to allow for the automatic submission / execution of this JCL, which means that NatQuery will be placing generated JCL streams directly into JES (for MVS systems) or POWER (for VSE systems) using automated FTP. This approach is referred to as **Direct FTP**. With **Direct FTP** – JCL is directly introduced directly into the mainframe server's Reader Queue from a NatQuery workstation - where it will be processed based upon a designated Job Class and the attributes of the initiator for that Job Class. In this way, fully automatic JCL execution can be achieved.

In other more isolated cases, NatQuery may not be allowed to directly access the JES or POWER Reader Queue via FTP, and instead will be configured to FTP NatQuery generated JCL into sequential files on the mainframe server. This approach is referred to as **Just FTP**, and using **Just FTP** essentially forces that manual execution will then occur - meaning that the JCL files are then placed into JES or POWER via manual intervention by a user.

If **Just FTP** is utilized by a site, then it is expected that users will know how to move / submit a NatQuery-generated JCL stream that has been placed onto the server via FTP into JES or POWER.

In general, a request submitted against a mainframe server will be a single file: A JCL file. In some cases more than one file will be moved, such as the case with a NatCDC execution which requires both a parameter files and a JCL file.

Execution of Scripts on non-Mainframes

When integrating against non-mainframe servers, NatQuery generates Script to accomplish a given process. NatQuery can then either move this Script onto the Natural server either by using automated FTP, or NatQuery can accomplish this by automating basic network copy commands across shared disk.

Once the Script is placed onto the Natural server, then NatQuery can automatically execute the Script by using any workstation-based command-line utility that allows for this. For UNIX / LINUX Natural servers, this would include commands such as RSH, REXEC, or other third-party software such as SSH. For Windows servers, this generally means the purchase or acquisition of third-party software such as PSEXEC (www.sysinternals.com). These approaches are referred to as **Direct FTP** (if FTP is used) or **Direct Copy** (if shared network disk is used).

One alternative to using NatQuery-initiated Command-Line commands (such as RSH, REXEC, SSH or PSEXEC) to initiate an execution of a Script placed on the server platform, another automated approach allows for these Scripts to be submitted / shelled by a server-based process. For example, NatQuery could be configured to place generated Scripts and supporting files into a specific target directory on the server using **Direct FTP** or **Direct Copy**. On the server platform, either a simple cron job, scheduled task or other third-party software (such as OPALIS) could then execute the request.

If manual integration is desired or otherwise warranted, then **Just FTP** or **Just Copy** will allow for this. In this situation, NatQuery will only place generated Scripts and supporting files onto the server platform into a specified directory where these processes would then be manually executed.

When NatQuery is used against non-mainframe systems, it should be noted that a single request is generally represented by a collection of files: A script (with a ".sh" or ".bat" extension), a natural program (a text file with an ".nsp" extension), and a CMOBJIN file CMSYNIN file (text files with ."txt" extensions). In some cases, additional files may also be moved – for example a NatCDC request – which will additionally move a parameter file.

Specific to UNIX / LINUX servers, execute permissions must allow for requests submitted by users to execute NatQuery-generated processes, or the command CHMOD can be used to make a given file executable (for example: "chmod 777").

If **Just FTP** or **Just Copy** is utilized by a given site, then it is expected that users will know how to move / submit a NatQuery-generated JCL stream that has been placed onto the server via automated FTP or Network Copy.

Creating Custom JCL / Script Templates

To extend the capability of NatQuery beyond the provided templates, NatQuery allows an Administrator to create custom JCL / Script templates which can then be associated with specific **Extract Type**(s) – such that any capability to handle data that is available in batch can be brought to bear on the data that NatQuery provides. For example: Custom SORT routines, File Encryption, invoking E-mails, automated file movement, or to introduce a Natural program into Natural on the server but save it instead of run it (or variations of this).

Custom JCL / Script templates are selected for use by an extract user at the point in time they have indicated they wish to perform **Send To Server** processing, and they have selected an **Extract Type** that the administrator has associated with a given custom JCL / Script template.

Ordinarily there is only a single JCL / Script template associated with a given **Extract Type**. This template will be seen on the **Send To Server** window within a combo box found on the right-hand side of the **Send To Server** window, in a frame labeled **Output Handling Options**, and it will be the only option to select in that combo box.

With a Custom JCL / Script template (and assuming that the administrator has both created a new template AND associated it to at least one **Extract Type**), a user will be able to over-ride the default **Output Handling Option** template by selecting the custom template from the combo box.

When creating a Custom Template, it is advisable to start with a template that is both working within NatQuery and is similar to the desired processing, however as a general rule of thumb the Production Request Process template is a good starting point.

To create a Custom JCL / Script template, the following steps would be performed:

1. **Start NatQuery if not already started**
   Information on starting NatQuery can be found in the section entitled Starting NatQuery.

2. **Invoke Administer JCL / Script function**
   On an empty NatQuery desktop, click on **Administer / Environment Configuration / Server Connection Configuration / JCL/Script Information**. These actions will invoke the **Administer JCL / Script** function.

   If the intent is to use an existing template as a base, then at this time it would be advisable to bring up the template that will be used as a base and then copy this template onto the Windows clipboard such that the template can be later pasted elsewhere. To do this, select the template of interest by locating the template's name in the combo box labeled **Server JCL/Script Component**. Once located, select this template by clicking on its name. This will load the template into the text box located in the center of the Administer JCL / Script window. Select all of the JCL / Script by high-lighting it. When all is highlighted (or at least that portion the Administrator intends to copy is highlighted), hold down both the “**Ctrl**” key, and the “**C**” key, and then release both. These actions will copy the highlighted portion of the base JCL / Script onto the Windows clipboard.

3. **Name New Template**
   While on the **Administer JCL / Script** window, click the **New** button. This action will invoke an Input Box that requests the name of the new template. To proceed to enter a new template the Administrator would input a unique template name, then click the **OK** button, and then proceed to the next step.

   If however the user wishes to **Cancel**; this can be accomplished by clicking the **Cancel** button – whereby the user will be returned to the Administer JCL / Script window.

4. **Create New Template**
   As a result of the above steps, the user should now be placed back on the Administer JCL / Script function window with the name of the template just entered being displayed in the combo box in the top right (labeled **Server JCL/Script Component**).

   The text box located in the center of the window will have a text message similar to:

   “This new JCL template contains no JCL/Script Information.”

   This entire text string should be highlighted, with the new JCL / Script template being entered into the central textbox.

   If a template was copied onto the windows clipboard, this template can be inserted into the central textbox by clicking anywhere in the central textbox, hold down the “**Ctrl**” key, tap the “**V**” button, and then release the “**Ctrl**” key. These actions will paste the contents of the clipboard into the central textbox.

   Whether the Administrator is creating a JCL / Script template from scratch or is using another template as a base, the JCL / Script template should be now be entered until it is thought to be complete, at which point the **Save** button should be clicked.

5. **Associate New Template to Extract Type(s)**
   With a new template created and displayed via the **Administer JCL / Script** window, the Administrator would then make an association between the newly created template and the **Extract Type**(s) that will be able to utilize the new script. This is accomplished by clicking the **Associate** button.

   Clicking the **Associate** button will invoke **Edit JCL/Script Association** window.

   The **Edit JCL/Script Association** window displays the available **Extract Type**(s) on the left, and allows for the individual **Extract Type**(s) to be associated with the newly created template. Specific Help on using the **Edit JCL/Script Association** window can be obtained by clicking the **Help** button while this function is displayed.

When all **Extract Type**(s) that should be associated with the newly created template show in the right-hand **Current Associations** frame, the Administrator will click the **OK** button. This action will return the user to the **Administer JCL / Script** window.

If other Custom templates are to be entered the Administrator may proceed using the information supplied above. As a general rule of thumb however, the creation of a Custom JCL / Script template should generally cause the Administrator to immediately test the new option to insure that there are no JCL / Script errors.

Handling JCL / Script Using External Editors

As opposed to using the basic JCL / Script editor provided by NatQuery, the Administrator may wish to use the text editor of their choice. This is possible because the JCL / Script templates used by NatQuery are stored as simple text files in an Environment Configuration path.

The only issue with using an external editor to edit JCL / Script templates using an external editor is that some NatQuery templates are considered "core" to the functioning of NatQuery. When changing these templates – it is recommended that subsequent to making the JCL change the Administrator should run a Verify Environment Configuration function against the Environment Path that contained the changed JCL / Script files.

The following table details the file naming conventions of each if the files, and additional information of interest to using an external editor to handle JCL / Script changes.

External Template Name	**Internal Name**	**Change Requires Verify**
Production Sequential Process	*Server-type***A.prd**	
Production ADABAS Process	*Server-type***B.prd**	
Production Predict Process	*Server-type***D.prd**	
Production E-Mail Process	*Server-type***E.prd**	
Production ADACMP Process	*Server-type***F.prd**	
Production ADAULD Process (direct)	*Server-type***G.prd**	
Production DWH Process	*Server-type***H.prd**	
Production ADAULD Process (Savetape)	*Server-type***I.prd**	
Production Summary Process	*Server-type***J.prd**	
Production DB2 Loader	*Server-type***L.prd**	
Production NatCDC Process	*Server-type***P.prd**	Yes
Production Request Process	*Server-type***R.prd**	Yes
Production Special Process (SYSTRANS)	*Server-type***U.prd**	
Production Server XML Process	*Server-type***X.prd**	

Note 1:
The value of *Server-type* italicized in the above table will be UNIX, MVS, VSE or NT depending upon which platform the Natural server resides on.

Note 2:
Administratively-created Custom JCL / Script templates generally adhere to the same naming conventions outlined above. These templates will be named with a prefix of Server-type of UNIX, MVS, VSE or NT (depending upon the platform that Natural resides on), will have a uniquely assigned numerical suffix (as opposed to the alphabetic suffix used with core templates), and will have an extension of ".prd" (to indicate a "production template – this as opposed to ".exm" which denotes an example template).

Natural Server Maintenance

Natural Server maintenance concerns the manipulation of information that is related to the remote Natural environment against which NatQuery will operate.

The type of changes that **Natural Server** maintenance covers are:

- **Natural Library**
  The **Natural Library** is a text string that will be dynamically substituted into JCL / Script references to &&USER-NATURAL-LIBRARY

- **Date Format**
  The Date Format designates the Edit Mask that NatQuery will use when outputting Date (D) fields.

- **Server Communication Mode**
  The **Server Communication Mode** dictates the method with which NatQuery will move objects between the workstation and the remote Natural platform.

- **Decimal Character**
  The **Decimal Character** designates the value that NatQuery will use when handling fields that require a decimal character. This value equates to the Natural environment’s setting of the DC parameter.

- **Variable Prefix Character**
  The **Variable Prefix Character** that will be used by NatQuery as a prefix for program variables generated into a Natural program. This value will default to the US pound sign, or “#”.

- **Natural Command Delimiter Character**
  The **Input Delimiter Character** that will be used when passing commands to Natural. This value equates to the setting of the Natural environment’s setting of the ID parameter.

- **Natural Security**
  The **Natural Security** checkbox is used to instruct NatQuery to prompt a user for a Natural Security User ID and Password, with NatQuery then dynamically substituting these user-entered values for &&NATSEC-USER-ID and &&NATSEC-USER-PWD that may exist in JCL / Script.

- **Multi-Fetch Options**
  The Multi-Fetch Options dictate how NatQuery will automatically invoke Multi-Fetch / Pre-Fetch (in situations where it is appropriate to do so).

The below instructions give a high-level view of how to access Natural Server Information. For a more in-depth discussion of the facets of **Natural Server** information, the reader may wish to refer to the section entitled **Error! Reference source not found.**.

To perform maintenance on Natural Server Information, the Administrator will utilize the **Administer Natural Server Information** function. To access this function, perform the following:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Information on how to start NatQuery is available in the section entitled Starting NatQuery.

2. **Invoke the Administer Natural Server Information function**
   With no open query on the NatQuery desktop, click on the **Administer** drop-down menu, then click on **Environment Configuration / Natural Server Information**. This will invoke the **Administer Natural Server Information** window.

3. **Perform Required Maintenance to Natural Server Information**
   On the **Administer Natural Server Information** window; make whatever changes are desired using the options presented.

   If help is required this can be found by clicking the **Help** button.

   When all desired changes are completed, click the **OK** button to close the **Administer Natural Server Information** window and apply the changes. As the window closes, NatQuery will notify you that the changes made require that the Verify Environment Configuration function should be run, and you will be asked if you wish to run this function.

   At this point in time, and due to the fact that **Natural Server Information** has changed – all extract generation against the current Environment Configuration will be disabled until the Verify Environment Configuration function has been executed.

   If the Administrator has additional changes that are required to be made against the current Environment Configuration, then in the interest of saving some time the Administrator can bypass the execution of the Verify Environment Configuration process by clicking the **No** button, make whatever other configuration changes are necessary, and then at a later point execute the Verify Environment Configuration function. As the time required to process the Verify function is not lengthy – it is suggested to run the Verify Environment Configuration function by clicking the **Yes** button when prompted to do so by NatQuery.

4. **Verify Environment Configuration**
   Once all **Natural Server Information** changes have been made, as well as any other Environment Configuration changes that may have been performed, the Administrator should execute the Verify Environment Configuration function.

   The Verify Environment Configuration function can also be executed manually by clicking on the **Administer** drop-down menu, then clicking on **Environment**

Configuration / Verify Environment Configuration.

When the Verify Environment Configuration function is executed, the Administrator should review the results of this report. If no unexpected problems are uncovered in this review, then the Administrator can continue to the next step. Otherwise, the Administrator should take any corrective action concerning difficulties that may exist, and then re-execute the Verify Environment Configuration function until the Administrator is satisfied with the results.

Further information on Verifying an Environment Configuration can be found in the section entitled Verifying an Environment Configuration.

5. **"Roll Out" the Environment Configuration (optional)**
Once the Environment Configuration has been verified, this configuration is ready for use, and may additionally be "rolled out" as desired. For information on "rolling out" a NatQuery Environment Configuration (I.E. making an Environment Configuration available to users), please refer to the section entitled Rolling Out an Environment Configuration.

Server Connection Maintenance

Server Connection maintenance within NatQuery is centralized into functions that are available under the **Server Connection Configuration** menu. This menu can be found by first clicking on the **Administer** drop down menu, then clicking on **Environment Configuration / Server Connection Configuration**.

There are four major functions that are available under the **Server Connection Configuration** menu:

- **Server Connection Information**
  The **Server Connection Information** function captures information that is specific to how NatQuery will interact with the remote Natural / ADABAS server.

  For information on handling **Server Connection Information**, it is suggested that the reader refer to the section entitled Step 2 - Server Connection Information as that section describes manipulating Server Connection information from the standpoint of a new install.

- **User Information**
  The User Information function is used to manipulate information relating to the valid users of a given NatQuery Environment Configuration.

  For information on handling User Information, the reader should refer to the section entitled User Maintenance.

- **JCL / Script Information**
  JCL / Script information handles the many ways that NatQuery will interact with the Natural server's batch environment.

  For information on handling JCL / Script Information., the reader should refer to the section entitled JCL / Script Templates.

- **Upload NatQuery Server Programs**
  The Upload NatQuery Server Programs function serves to perform an upload of the Natural programs that NatQuery will interact with on the server platform.

  This processing is usually intended to only be used once as described in the section entitled Step 5 - Server Platform Initialization.

Depending upon the modifications required, the Administrator should refer to one or more of the references shown above to implement changes to **Server Connection Information**.

Verifying an Environment Configuration

When changes are made to an Environment Configuration, NatQuery typically requires that the **Verify Environment Configuration** function be executed.

This section outlines the processing behind the **Verify Environment Configuration** function to assist the reader in better understanding this processing. For information specific to interpreting a Verification Report, please refer to the section entitled Interpreting and Reacting To A Verification Report.

One of the primary roles of the **Verify Environment Configuration** function is to check for the existence of basic information that is required to allow generation against any given source DDM. Examples of basic DDM edits would be:

- DDMs which have no **Descriptor Statistic Information** whatsoever (and the DDM does not represent a sequential file);

- DDMs which have either MU fields and or PE fields against which no **Occurrence Information** has been entered.

The **Verify Environment Configuration** process also reviews the integration / connectivity information. Example of this type of editing would include:

- FTP integration has been indicated however portions of required FTP information are not present or appear invalid

- The existence of "core" JCL / Script templates which are central to NatQuery's ability to execute requests on the Natural server platform are missing.

Beyond its role of attempting to insure that proper configuration data is present and appears to be entered correctly, the **Verify Environment Configuration** function serves another role: It attempts to insure that key configuration files are not changed in an unauthorized manner. This is done by the internal creation of an encrypted file called **VALIDENV.CFG** that is generated as a result of the **Verify Environment Configuration** process.

As the **Verify Environment Configuration** function executes and edits for specific facets of information; this editing is done against data captured from files that are built as a result of the various configuration steps. As the data from these files are edited, the file VALIDENV.CFG is created that records the file name and additionally captures the current date and time stamp of the file at the time of the verification process. When all file information is captured, then VALIDENV.CFG is encrypted and placed into the current Environment Configuration path.

When a version of NatQuery starts, one of the first things it looks for is the existence of the VALIDENV.CFG file in the currently designated Environment Configuration path. If this file is not found then NatQuery knows that the current Environment Configuration has not been verified or does not exist in the path specified and will produce a message to this effect. In this situation all extract generation will be disabled, and depending on what other information is missing and/or invalid, other functions may be disabled as well.

If NatQuery is started and it does find the VALIDENV.CFG file, NatQuery will then internally decrypt the contents of this file which yields the date and time stamps of the key Environment Configuration files at the time they were last verified. NatQuery then compares this information to the current date and time stamps for these same files. If all date and time stamps match (and assuming that a fully configured Environment Configuration did exist previously) then the corresponding functions that rely on that information will become available. If any date and time stamp does not match, then NatQuery will examine the file(s) which no longer match and will disable those functions that would otherwise utilize the information from those files.

In this manner, it becomes difficult for an Administratively provided Environment Configuration to be manually "manipulated" outside of NatQuery because at the time a configuration-related file is changed - its date and time stamp would also change - and NatQuery would detect this change.

To manually invoke the **Verify Environment Configuration** process, the Administrator would perform the following:

1. **Start NatQuery if not already Started**
   Information on Starting NatQuery can be found in the section entitled Starting NatQuery.

2. **Invoke Verify Environment Configuration function**
   On an empty NatQuery desktop, the administrator would click the **Administer** drop-down menu, then click **Environment Configuration / Verify Environment Configuration**. This action will internally invoke the Verification processing, and at the end of this processing a **Verification Report** will be produced in a window.

3. **Review Verification Configuration Report**
   The verification process will result in the creation of a report that details virtually all aspects of an Environment Configuration.

   In essence, the **Verification Report** serves to provide an Administrator an "action-item" list that describes the status of the **Environment Configuration**, and details and deficiencies that may exist within this **Environment Configuration**. Using the details of this **Verification Report**, the Administrator can then take the indicated actions to correct the deficiencies. Further information on the types of issues that a **Verification Report** might detail is discussed below.

   When the review of the **Verification Report** is complete (this report can also be printed if required), then **Verification Report** can be closed by clicking the **OK** button. Clicking the **OK** button will close the **Verification Report** window, and will return the user to the NatQuery desktop.

Interpreting and Reacting To A Verification Report

A **Verification Report** essentially is broken down into three sections:

- Basic Information Section**,**
- DDM Verification Section, and
- Server Communication Section.

Basic Information Section

The **Basic Information Section** details basic information about the version of NatQuery being executed, the date and time of the Verification, the path of the current Environment Configuration, the date and time of the "core" configuration files in that Environment Configuration path. These "core" configurations files would include the file that stores **File Relationship Information** (FILERELS.CFG), the file that stores the **Descriptor Statistics Information** (FILEDESC.CFG), the file that stores **Occurrence Information** (FILEOCCS.CFG), and the file that stores **Server Information** (SERVER.CFG).

If the Administrator discovers that the **Environment Configuration Path** being reported on (as seen in the report text that relates to Configuration File Path) is not the correct path, then the Administrator should change the current **Environment Configuration Path** to be the correct path and then re-run the **Verify Environment Configuration** function. The value of the **Environment Configuration Path** can be changed by using the **NatQuery Configuration** function. On an empty NatQuery desktop, this function is accessed by clicking on **Administer** drop-down menu, then clicking on **NatQuery Configuration**, and then clicking on the **Environment Paths** tab.

In the final portion of the **Basic Information Section**, NatQuery will display the current value of **Target Library** (this would be the value of the **Natural Library** that will be substituted into JCL / Script references against the **Dynamic Substitution Variable** &&NATURAL-LIBRARY), and the current **Communication Mode** (in most cases this will be "**FTP"**, but this could also be "**PC Network**" or "**none**").

To change either the value of **Target Library** or **Communication Mode**, the Administrator would use the **Administer Natural Server Information** function. On an empty NatQuery desktop, this function would be accessed by the **Administer** drop-down menu, then clicking on **Environment Configuration**, and then clicking on **Natural Server Information**.

DDM Verification Section

NatQuery examines each DDM that it finds in the current **Environment Configuration** path and the Verification Report will present the results of this examination in alphabetical sequence by DDM Name. If a particular DDM of interest is not found in the Verification Report – then it does not exist in the current **Environment Configuration** path.

To **Add** a new DDM into the **Environment Configuration Path**, the Administrator should refer to the section entitled Adding a New DDM.

To **Modify** an existing DDM that resides in an **Environment Configuration Path**, the Administrator should refer to the section entitled Modifying An Existing DDM; to **Delete** an existing DDM the section entitled Deleting An Existing DDM should be referred to.

For each DDM found in the Environment Path, the Verification Report will produce the following tag:

```
FILE-NAME (DDMxxx.nsd)
```

The ***FILE-NAME*** tag will be the external name of the DDM. The **DDM*xxx*.nsd** will be the internal name of the file in the Environment Configuration Path that represents the DDM, with the "***xxx***" portion being a unique NatQuery-assigned number.

For each DDM, the following four things will be examined:

1. **Sequential DDM Check**
   For each DDM found in the **Environment Configuration Path** , NatQuery will check to see if the DDM represents a Sequential File. In NatQuery syntax this means that the DDM will have a value of "**SEQ**" for both the DB (DBID) and FILE (FNR) values in the DDM header line.

   Whether or not a DDM represents a Sequential File will come into play for editing **Occurrence Information** or **Descriptor Statistic** information for that specific DDM (since neither should apply to a true sequential file). **File Relationship Information** editing/ reporting may occur still as detailed below.

2. **Recurring Fields Edits (non-Sequential Files Only)**
   For each DDM found in the **Environment Configuration Path** that does not represent a Sequential File, NatQuery will examine the DDM for the existence of any recurring fields such as Multi-Valued fields (MUs), Periodic-Group fields (PEs), or MUs in PEs.

   The results of this examination will be shown in the report under the tag:

   ```
   - Checking Occurrence Information
   ```

   If a DDM contains no recurring fields, then the DDM will not require any **Occurrence Information**, and NatQuery will report this fact with a message similar to:

```
* No Occurrence Information is required for DDM
```

If the above message is displayed for a DDM, then it does not require any further action in regards to **Occurrence Information**.

If a DDM does contain recurring fields, then the DDM will require **Occurrence Information**, but this information may already exist and be correct. If this is the case, then the Verification Report will reflect this with a message similar to:

```
* Occurrence Information is valid
```

If a DDM contains recurring fields and NatQuery finds that no **Occurrence Information** exists, then the Verification Report will reflect this with a message similar to:

```
* Occurrence Information is required but does not exist
```

In a similar fashion, if a DDM contains recurring fields and NatQuery finds that while **Occurrence Information** does exist for some fields but is missing for others, then the Verification Report will reflect this with a message similar to:

```
* No Occurrence Information exists for field-name
```

Either of the above two messages would be rectified by manipulating **Occurrence Information** for the specific file, which is accomplished through the **Administer Occurrence Information** function.

To determine **Occurrence Information** for all recurring field that may exist in a file, then the Administrator would use the **Data Discovery** function called **Repeating Field Analysis**.

To manually enter or manipulate **Occurrence Information**, or to **Import** data from a previously-executed **Repeating Field Analysis**, the Administrator would use the **Administer Occurrence Information** function. This function is accessed from an empty NatQuery desktop by clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration / Data Discovery / Analysis / Repeating Field Analysis**.

For detailed information on handling **Occurrence Information** for a DDM, the Administrator should refer to the section entitled Occurrence Information.

3. **Descriptor Statistic Edits (non-Sequential Files Only)**
For each DDM found in the **Environment Configuration Path** that does not represent a Sequential File, NatQuery will examine the DDM for the existence of any Descriptors, Sub-Descriptors or Super-Descriptors.

The results of the examination will be shown in the report under the tag:

```
  - Checking Descriptor Statistics
```

If a given DDM represents a sequential file, then the report will reflect this with text similar to the following:

```
    * Descriptor Statistics not required for sequential file
```

If a given DDM does not represent a sequential file and **Descriptor Statistic Information** exists (even if “zeroed out”), then the report will reflect this with text similar to:

```
    * Descriptor Statistics are valid
```

If a DDM contains Descriptors, Sub-Descriptors or Super-Descriptors and NatQuery finds that no **Descriptor Statistic Information** exists, then the Verification Report will reflect this with a message similar to:

```
    * Descriptor Statistic Information are required but do not exist
```

In a similar fashion, if a DDM contains Descriptors, Sub-Descriptors or Super-Descriptors and NatQuery finds that while **Descriptor Statistic Information** does exist for some fields but is missing for others, then the Verification Report will reflect this with a message similar to:

```
    * Descriptor Statistics for field-name do not exist
```

Finally, the situation may arise where Descriptor Statistic Information exists for a field that was previously known as a Descriptor, Sub-Descriptor or Super-Descriptor; but this designation has subsequently changed in the DDM. This situation would be reflected in the report with text similar to:

```
    * Descriptor Statistics reference field-name which is not in DDM
```

Any one of the last 3 messages would be rectified by manipulating **Descriptor Statistic Information** for the specific file, which is accomplished through the **Administer Descriptor Statistic Information** function.

To determine **Descriptor Statistic Information** for all Descriptors, Sub-Descriptors or Super-Descriptors that may exist in a file, then the Administrator would use the **Generate** button after having selected a specific file using the **Administer Descriptor Statistic Information** function**.**

To manually enter or manipulate Occurrence Information, or to **Import** data from a previously-executed **Descriptor Statistic Analysis**, the Administrator would use the **Administer Descriptor Statistic Information** function. This function is accessed from an empty NatQuery desktop by clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration / Descriptor Statistics**.

For detailed information on handling **Descriptor Statistic Information** for a DDM, the Administrator should refer to the section entitled Descriptor Statistic Information.

NOTE:
If NatQuery is being used primarily as a data extraction tool for Data Warehousing efforts, then detailed Descriptor Statistic Information is not a requirement and can be "zeroed out".

4. **File Relationship Information**
Any existing **File Relationship** information previously entered against a given DDM will be displayed in the report following the **Descriptor Statistic** edits.

File Relationship details for a given DDM will be shown in the report under the tag:

```
   - Reviewing File Relationships
```

Under this tag, NatQuery will examine any **File Relationships** that relate to the given DDM, either from the given DDM into another DDM, or from another DDM into the given DDM.

If the given DDM has no File Relationships that point to any other DDMs, then this will be reflected in the report with text similar to:

```
     * This file has no relationships to other files
```

If a specific DDM has one or more **File Relationships** defined to other DDMs, then these relationships will be individually detailed in the report under the tag:

```
     * File Relationships to other Target Files
```

If the given DDM is the target of a File Relationship from another file, then any such **File Relationships** will be individually detailed in the report under the tag:

```
     * File Relationships to Current File
```

If the given DDM is not pointed to by any other defined DDM **File Relationship**, then this will be reflected in the report with text similar to:

```
     * There are no files with a relationship to this file
```

For any **File Relationship** that does exists for a given DDM, the **Verification Report** will display the Source File, the Target File, and the Cardinality of the relationship – and will additionally display the Foreign or Primary Key that is used to implement the File Relationship.

Any File Relationship information that is not defined and which should exist for

NatQuery to function properly would be created by using the **Administer File Relationship** function. Similarly, the Administer File Relationship function would also be used to modify or delete File Relationship Information as needed. This function is accessed from an empty NatQuery desktop by clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration / File Relationships**.

For detailed information on handling **File Relationship Information** for a DDM, the Administrator should refer to the section entitled File Relationship Information

After all DDM verification results are presented in the **Verification Report**, the **Verification Report** will produce a Summary of the DDMs found in the current Environment Configuration Path. This summary will show the number of DDMs reviewed by the verification process, the number of these DDMs that are Verified for use (meaning that Occurrence Information if needed exists and appears correct, and that Descriptor Statistics also exist and appears correct), and will additionally detail the number of DDMs (if any) that failed verification.

Server Communication Section

The final section of a Verification Report will detail the general status of how NatQuery will communicate with the Natural Server.

The first portion of the Server Communication Section will provide the Date and Timestamps of the two internal files that record basic Server Communication information: The Server Communication file itself (SERVER.CFG), as well as the file that contains the Defined Users of NatQuery (FTPUSERS.CFG).

If NatQuery has been configured to have a Server Communication Mode of “none”, or the internal Server Communication configuration file (SERVER.CFG) cannot be found in the current Environment path, then the Verification report will report this with a text message similar to:

```
* No Server Communication will be performed by NatQuery
```

Typically, the **Server Communication Mode** for NatQuery is set to “**FTP**”, with this mode being set through the **Natural Server Information** function. This function is accessed on an empty NatQuery desktop by first clicking on **Administer**, then clicking on **Environment Configuration / Natural Server Information**.

The next portion of the **Verification Report** details any Production versions of the standard JCL / Script templates that may exist in the current Environment Configuration Path.

While there are many types of standard JCL / Script templates that NatQuery can recognize and manage, NatQuery itself only requires the Production Request Process JCL / Script template to exist in order to allow a Verification to succeed, and it is generally recommended that the Production Special Process JCL / Script template be available as well (this is the template used for handling the download of DDMs / FDTs. For further information on handling JCL / Scripts, it is suggested that the administrator refer to the section entitled JCL / Script Templates.

Following the JCL / Script templates, the verification report will provide information on the **Server Connection**, with this information being managed through the **Natural Server Information** and **Server Connection Configuration** functions.

Following this, the **Verification Report** will display basic details about any defined users of NatQuery, with the information displayed showing the defined User Ids, the corresponding User Names, the relative status of the user, as well as the number of request slots that each user has defined. For further information on handling / defining a user of NatQuery, the administrator should refer to the section entitled User Maintenance.

At the very end of the **Verification Report**, NatQuery will display the overall status of the selected **Server Communication Mode**.

When using **FTP** as a **Server Communication Mode**, and assuming that the Server Configuration File (SERVER.CFG), the NatQuery User File (FTPUSERS.CFG), and the

Production Request Process JCL / Script template all exist in the current Environment Configuration Path: Then NatQuery should report this status with a text similar to the following:

```
* FTP Mode is verified for use
```

If the Verification process detects errors with the designated Server Communication mode, this will be reflected in the report with a message similar to:

```
* FTP Mode is not verified for use
```

If the above message is displayed, then the reason(s) for the Mode not being Verified will be detailed, and the Administrator should then take the corrective action necessary to resolve the error as detailed above.

The Administrator will therefore use the Verify Environment function to not only pinpoint errors that may exist in an Environment Configuration, but also to "certify" (verify) an Environment Configuration for use by users.

By using the information provided in the report; the Administrator may be guided on what changes need to be made, if any, for NatQuery to be used as intended and expected.

Enabling E-Mail handling of Extracted Output

When a user typically requests a data extract with NatQuery, NatQuery generates a Natural program that will resolve the request. This Natural program is then imbedded into the Production Request Process JCL / Script template, and then this JCL / Script template is at some point executed in batch on the Natural server platform. When this request has completed execution, the extract data created by the extraction program becomes available to the user through a user-initiated download.

As an alternative to having the user monitor a given request's status, and then initiate a download of the extracted data, it is usually possible to configure NatQuery so that extracted data can be automatically sent as an e-mail attachment to the initiating user (or to another single e-mail address) through the execution of the data extraction process itself.

This e-mail functionality is implemented by enhancing the **Production Request Process** JCL / Script template so that it has an additional "job step". This additional job step would be executed immediately after the batch Natural job step executes (the job step that executes the generated Natural extract program creating the required data into a named file), and this additional step would then execute an existing server-based utility program that can accept a server-based file name as an argument and create an e-mail that will contain the named file's data as an attachment.

This functionality would become available to an End-User at the point that they use the **Send To Server** function and they then select "**Download To PC File**". With "**Download to PC File**" selected, the user will be able to select "**Production E-Mail Process**" as an Output Handling Option (this will normally default to "**Production Request Process**").

To enable this feature, the Administrator should begin by investigating the functionality of existing server-based utility programs that would enable this ability. When called in or through batch, this utility program must be able to function using parameters that are either supplied through existing NatQuery dynamic substitution variables, or which can be hard-coded into the execution JCL / Script template itself.

In most situations, it is envisioned that the dynamic substitution variables that NatQuery provides will be adequate to provide the existing server-based e-mail utility program with the parameters needed to enable the e-mail sending of extracts as attachments. In those situations where NatQuery does not provide the required processing, NatWorks, Inc would like to hear from you so that we might jointly investigate available alternatives. Information on contacting NatWorks can be found in the section entitled Contacting NatWorks.

Once it has been determined that the parameters that need to be provided to the server-based e-mail utility can be provided by NatQuery, then the following steps will enable this capability:

1. **Start NatQuery**
   Start NatQuery if it is not already started. Instructions on how to start NatQuery can be

found by referring to the section entitled Starting NatQuery.

2. **Enable E-Mail Handling**
Invoke the **Administer Server Connection Information** function by clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration / Server Connection Information / Server Information**. This will invoke the **Administer Server Connection Information** window.

   With the **Administer Server Connection Information** window presented, click on the **Miscellaneous** tab.

   On the **Miscellaneous** tab, select the option entitled **Enabled E-Mail Handling of Output Data**. Enabling this will have three effects: It will enable a graphical control on the NatQuery **Send to Server** window that will allow a user to designate e-mail handling, it will further allow for the creation of a JCL / Script template named Production E-Mail Process, and it will make available several dynamic substitution variables within the Administer JCL / Script function to support E-mail operations.

   Click the **OK** button to close the **Administer Server Information** window. As this window closes, NatQuery will present a message box indicating that the Verify Environment Configuration function should be run, and will ask if you want to run this. It is suggested that you answer **No** to this prompt at this time, as you will have to run the Verify Environment Configuration function after modifying the Production Request Process JCL /Script (see the next step of this section).

3. **Create the Production E-Mail Process JCL / Script Template**
Since the functioning of the **Production E-mail Process** will be generally based on the functionality provided by the **Production Request Process** JCL / Script template – the best way to create the **Production E-mail Process** JCL / Script is to base this template upon the **Production Request Process** template.

   Since both templates are managed through the **Administer JCL / Script** function, the administrator should begin by accessing this function. The **Administer JCL / Script** function is invoked on an empty NatQuery desktop by clicking on **Administer / Environment Configuration / Server Connection Configuration / JCL / Script Information**.

   Once in the **Administer JCL / Script** function, the administrator will select the **Production Request Process** template by using the combo box in the upper right hand corner. Selecting this template will load the JCL / Script for this template into the central textbox of the **Administer JCL / Script** window. This template should be copied onto the Windows clipboard by highlighting the displayed template in its entirety and then click and hold the "**Ctrl**" key and tap the "**C**" key, then release both keys. These actions will copy the highlighted text onto the internal Windows clipboard.

   The administrator would now click the **Production E-Mail Process** JCL / Script template by using the combo box in the top-right corner of the window.

If the **Production E-Mail Process** JCL / Script did not previously exist in the Environment Configuration path, then selecting the **Production E-Mail Process** will re-build the central text box so that it displays the following:

```
This template does not contain any JCL/Script information.
```

If this is the case, then this text should be replaced with the contents of the Windows clipboard by first highlighting this text, and then hold down the "**Ctrl**" key and tap the "**V**" key, then release the "**Ctrl**" key. This action will insert the content of the Windows clipboard into the textbox (which should be a copy of the **Production Request Process** JCL / Script).

With either a pre-existing or just-copied JCL / Script template now displayed in the central textbox, the Administrator will make whatever changes are necessary so that, the last step of the JCL / Script will take the data file created by the NatQuery-generated Natural program and handles this file through an SMTP program or similar program / application.

To assist in the passing of meaningful parameters to a E-Mail program, when the Enable E-mail handling option is selected, NatQuery will provide the following dynamic substitution variables:

&&USER-EMAIL-ADDRESS
&&USER-EMAIL-DIR-REF
&&USER-EMAIL-FILE-REF
&&USER-EMAIL-MESSAGE
&&USER-EMAIL-SUBJECT

For further information on these dynamic substitution variables, please refer to the section entitled NatQuery Dynamic Substitution Variable Reference Table.

4. **Save Production E-Mail Process Template**
   When all necessary changes or corrections have been made to the **Production E-Mail Process** JCL / Script template, the Administrator will click the **Save** button to save this template to a disk file in the Environment Configuration Path.

   The Administrator may then exit the **Administer JCL / Script** function by clicking the **OK** button.

Creating or modifying only the **Production E-Mail Process** JCL / Script template will by itself not require a Verification process. If change to the **Production E-Mail Process** occurs in conjunction with other JCL / Script templates that are also "core" templates – NatQuery may require a Verification Process, and will prompt for this to occur automatically.

If the **Production E-Mail Process** JCL /.Script process is just being created, it is advisable that the administrator now test the **Production E-Mail Process** by actually submitting a request, manually review the results of the batch execution, and also insure the correct reception of e-mail results. For assistance in resolving common problems that may arise with the JCL / Script's execution, the Administrator may wish to refer to the section entitled Troubleshooting.

Rolling Out an Environment Configuration

To allow for an Environment Configuration to be usable to any other NatQuery workstation, an Environment Configuration must be made available to these NatQuery installations.

For installations where there will only be a single installation of NatQuery, then there is no need to consider “rolling out” an Environment Configuration. This is because the Environment Configuration should already be available to the workstation upon which the given Environment Configuration was built, so single installations of NatQuery need not concern themselves with Rolling Out an Environment Configuration.

In situations where multiple installations of NatQuery will exist and a new Environment Configuration is built, this configuration needs to be provided to the NatQuery installations that require this configuration.

There are basically two methods of providing an Environment Configuration to other NatQuery installations. With both approaches, the Administrator would first build the required Environment Configuration on a single machine using an Administrator version of NatQuery.

Once built and verified, the Administrator could then use either of the following rollout approaches:

- **Networked Environment**
  With this approach, an Administrator builds a required Environment Configuration on a single machine using an Administrator version of NatQuery.

  Once the Environment Configuration is built and verified, the Administrator would place a copy of the Environment Configuration into a network path that is available to the multiple users of NatQuery.

  These users must be defined to have both Read and Write Access to this network directory.

  Each user of NatQuery using the network path will now be pointing (provide users with a new mapped drive if this is a new directory or new install) at the revised Environment Configuration.

  To utilize this approach, please refer to the section below entitled Rolling Out a Configuration in Networked Environments.

- **Non-Networked Environment**
  With this approach, an Administrator builds a required Environment Configuration on a single machine using an Administrator version of NatQuery, and then copies the Environment Configuration onto a media that can then allow the Environment Configuration files to be copied into a directory on these other non-networked machines.

To utilize this approach, please refer to the section below entitled Rolling Out a Configuration in non-Networked Environments.

When rolling out an Environment Configuration, it should be noted that it is possible to build multiple unique Environment Configurations, say for different ADABAS files / applications, with these unique Environment Configurations being kept in separate directories. For further information on handling multiple Environment Configurations, please refer to the section entitled Supporting Multiple Environment Configurations.

Rolling Out a Configuration in Networked Environments

In networked environments, the process of providing an Environment Configuration is simplified due to the network's ability to share file resources. Essentially, an Environment Configuration is built on one machine, and then this configuration is copied / export into a network directory against which defined users will have both Read and Write access.

To accomplish a roll out on a networked environment, perform the following:

1. **Insure Environment Path is Correct**
   Insure that the current setting of the Environment Path points to the path that contains the correct Environment Information. During the initial installation of NatQuery, this path is suggested to be the **Files** sub-directory of the NatQuery Installation directory.

   The current setting of the Environment Path can be seen on the NatQuery Toolbar.

   If the path setting is not correct, then click on the **Administer** drop-down menu, and then click on **NatQuery Configuration**. This will invoke the **NatQuery Configuration** window. On this window, click on the **Environment Paths** tab. On the **Environment Paths** tab, the contents of the text field named **Environment Path** represent the current setting. Using the **Browse** button associated with the path value, navigate to the appropriate path that contains the new Environment Configuration information. Close the **NatQuery Configuration** window by clicking on the **OK** button.

2. **Invoke Export Environment Configuration function**
   Click on the **Administer** drop-down menu, then click on **Environment Configuration**, and then click on **Export Environment Configuration**. This will invoke the **Export Environment Configuration** window.

   If the directory into which the newly built configuration will be exported already exists, then you may proceed to the next step.

   If the directory does not yet exist, then this directory can be created by clicking on the **Create Folder** button located on the **Export Environment Configuration** window. Clicking the **Create Folder** button will invoke the **Create New Folder** window that will allow the new directory to be created in the appropriate path on the networked drive. Once the directory has been created, then you can proceed to the next step.

3. **Export the Environment Configuration**
   Using the **Export Environment Configuration** window, insure that the **Export to Path** and the **Export to Drive** controls point at the correct networked directory. The actual export can then be initiated by clicking on the **Export** button, this action will export all Configuration Files from the current Environment Configuration Path to the designated path.

   If the directory that is being exported to existed previously and it contained a previous

Environment Configuration, NatQuery will now display a message box that will request permission to delete the older versions of the Configuration files. It is strongly suggested that the user respond to this prompt by clicking the **Yes** button.

Specific Help on Exporting an Environment Configuration can be found by clicking the **Help** button while in this function.

4. **Switch to Export Environment Path**
   Subsequent to the export completing, the Administrator will be returned on the NatQuery desktop. It is now strongly suggested that the Administrator test the freshly exported Environment Configuration by changing the current NatQuery Environment Configuration path to be that of the freshly exported directory.

   To accomplish this, the Administrator will click on the **Administer** drop-down menu, then click on **NatQuery Configuration**. These actions will display the **NatQuery Configuration** window, with the **User Identification** tab displayed.

   Click on the **Environment Paths** tab; this tab will allow for the modification of the installation's current **Environment Path** setting to be easily changed to the freshly exported directory. Once the new **Environment Path** has been set to the new value, the Administrator should close the NatQuery Configuration window.

   <u>Please note</u>: NatQuery may respond with any unexpected error relating to the just-exported Environment Configuration, this situation can be created due to Windows changing the date timestamps of files as they were copied. To correct this situation the Administrator should immediately run the **Verify Configuration** function. On an empty NatQuery desktop, this function would be accessed by first clicking on the **Administer** drop-down menu, and then clicking on **Environment Configuration / Verify Environment Configuration**.

   <u>Please note</u>: In the event that NatQuery responds with an error message during this process, until the Administrator has run the Verify Configuration process outlined above, this directory will be unusable to those users pointing at this path. Under normal operating circumstances the Verify Configuration process will take less then a minute to complete.

With the Environment Configuration fully tested on a networked drive, other NatQuery installations that are present on different workstations that can connect to this networked drive can now use this path in the Environment Path setting on their respective machines. In some cases where UNC naming is not utilized, this may require that specific network drives be mapped from a NatQuery installation workstation back to the file server's directory location.

Rolling Out a Configuration in non-Networked Environments

In non-networked environments, the “sneaker-net” approach works best. With this approach, an Environment Configuration is exported to transportable media by the Administrator, and then this transportable media is provided to other NatQuery installation for importation. To accomplish this, perform the following:

1. **Insure Environment Path is Correct**
   Insure that the current setting of the Environment Path points to the path that contains the newly built or newly updated Configuration. During the initial installation of NatQuery, this path is suggested to be the **Files** sub-directory of the NatQuery Installation directory.

   The current setting of the Environment Path can be seen on the NatQuery Toolbar.

   If the path setting is not correct, then click on the **Administer** drop-down menu, and then click on **NatQuery Configuration**. This will invoke the **NatQuery Configuration** window. On this window, click on the **Environment Paths** tab. On the **Environment Paths** tab, the contents of the text field named **Environment Path** represent the current setting. Using the Browse button associated with the path value, navigate to the appropriate path that contains the new Environment Configuration information. Close the **NatQuery Configuration** window by clicking on the **OK** button.

2. **Invoke Export Environment Configuration function**
   Click on the **Administer** drop-down menu, then click on **Environment Configuration**, and then click on **Export Environment Configuration**. This will invoke the **Export Environment Configuration** window.

   If the directory into which the newly built configuration will be exported already exists, then you may proceed to the next step.

   If the directory does not yet exist, then this directory can be created by clicking on the **Create Folder** button located on the **Export Environment Configuration** window. Clicking the **Create Folder** button will invoke the **Create New Folder** window that will allow the new directory to be created in the appropriate path on the networked drive. Once the directory has been created, then you can proceed to the next step.

3. **Export the Environment Configuration**
   Using the **Export Environment Configuration** window, insure that the **Export to Path** and the **Export to Drive** controls point at the correct transportable media. The actual export can then be initiated by clicking on the **Export** button, this action will export all Configuration Files from the current Environment Configuration Path to the designated media / directory.

   If the directory that is being exported to existed previously and it contained a previous Environment Configuration, NatQuery will now display a message box that will request permission to delete the older versions of the Configuration files. It is strongly suggested that the user respond to this prompt by clicking the **Yes** button.

Specific Help on Exporting an Environment Configuration can be found by clicking the **Help** button while in this function.

4. **Use Import function to Import into a separate NatQuery Installation**
On workstations that require the Environment Configuration, the **Import Environment Configuration** function can then be used against the transportable media to **Import** the required information. The **Import** function can be found by clicking on **Administer / Environment Configuration / Import Environment Configuration**.

   Specific Help on Importing an Environment Configuration can be found by clicking the **Help** button while in this function.

   New installations should be thoroughly tested prior to use interaction.

Supporting Multiple Environment Configurations

In many situations, a NatQuery Administrator may be required to support multiple groups of end-users, with each grouping of end-users having different data extraction needs (I.E. access to differing ADABAS files and possibly different environments). This then means that the Administrator will then build and manage multiple Environment Configurations, with each individual Environment Configuration being built into different directory paths.

As a review, an Environment Configuration is in actuality a collection of administratively built files that are stored into a specified workstation directory. For all versions of NatQuery (Administrative, End-User or Demo) the path from which NatQuery will resolve the Environment Configuration is determined by the setting of **Environment Path**, a field located on the **Environment Paths** tab of the **NatQuery Configuration** function.

For End-Users and Demo versions of NatQuery, the setting of **Environment Path** directly controls the Environment Configuration against which the End-User can execute data extractions. For Administrative versions this is also true, however: In Administrative versions the setting of **Environment Path** also controls the Environment Configuration that Administrative functions will manipulate.

When an Administrator begins the process of configuring NatQuery to handle the data extractions needs for the initial group of End-Users, one of the first steps in the Initial Configuration Process is the establishment of a directory path that will be used to contain the Environment Configuration that will be specific to needs of the initial group of End-Users. For this initial configuration of NatQuery it is suggested to use the **FILES** sub-directory of the **NatQuery** installation directory as the location in which to build the first Environment Configuration.

At the point in time that the Administrator is required to build additional Environment Configurations, the Administrator should begin this process by thinking about a suitable naming convention for the workstation directories on their local machine that will then be utilized to contain the separate and unique Environment Configurations.

With a naming convention established, the Administrator can then implement the naming convention by creating the necessary directories. With the required directories established, these directories can then be utilized to contain the required individual Environment Configurations.

To facilitate the creation of the required directories, the Administrator can make use of the NatQuery **Create Folder** function that is available either through the **NatQuery Configuration** window or the **Export Environment Configuration** window. Alternatively, the Administrator can make use of workstation tools and applications such as Windows Explorer to create the required directories.

To facilitate the movement of Environment Configuration information between directories, the Administrator can make use of the **Import Environment Configuration** window or the **Export Environment Configuration** window. Alternatively, the Administrator can make use of workstation tools and applications such as Windows Explorer to move / copy the contents of one directory to another directory.

As a suggestion to speed the Administrator through the process of building one or more subsequent Environment Configurations, the Administrator can consider making use of existing components of one Environment Configuration as a basis of another Environment Configuration. For example, it may be that while DDMs will be different between two groups, the **Natural Server Information**, **Environment Connection Information** and the **JCL / Script** templates may be virtually identical.

To build an Environment Configuration that is based on information already existing in another Environment Configuration, the Administrator could utilize the following approach:

1. **Invoke the NatQuery Configuration function**
   On an empty NatQuery desktop, enter the **NatQuery Configuration** function by clicking on **Administer** and then clicking on **NatQuery Configuration**. When the **NatQuery Configuration** function is presented, click on the **Environment Paths** tab.

2. **Create a New Directory**
   While on the **Environments Paths** tab of the **NatQuery Configuration** function, click on the **Create Folder** button. This action will invoke the **Create New Folder** window.

   On the **Create New Folder** window, provide a descriptive name for the new Environment Configuration, and designate a drive and directory that will contain the new folder. For organizational purposes, it is strongly suggested that the new folder be placed into the NatQuery installation directory (but this is not a requirement).

   Once the new folder has been named and its location designated, clicking the **Create** button will create the required folder and will then close the **Create New Folder** window, returning the user to the **NatQuery Configuration** window.

3. **Change Environment Configuration Path to New Directory**
   Click on the **Browse** button associated (next to) the **Environment Paths** text field. This action will invoke the **Select Environment Path** window. Using the controls of this window, navigate to and then select (double-click on) the folder you just created in step 2. With the new folder selected, you can now close the **Select Environment Path** window by clicking the **OK** button. This action will close the **Select Environment Path** window and will return the user to the **NatQuery Configuration** window.

4. **Close NatQuery Configuration window**
   You may now close the **NatQuery Configuration** window by clicking the **OK** button. As the window closes, NatQuery will provide warning messages indicating that the current Environment Configuration is either missing or non-existent. For the moment, you can ignore this error message, as the newly set Environment Configuration path (the newly created directory in step 2) is in fact currently empty.

5. **Invoke the Import Environment Configuration function**
   Click on **Administer**, then click on **Environment Configuration**, and then click on **Import Environment Configuration**. This action will invoke the **Import Environment Configuration** window.

6. **Import the "basic" Environment Configuration**
   Using the controls on the Import Environment Configuration window, point the **Import From Drive** and **Import From Path** controls so that the "basic" directory is referenced and then click the Import button. This action will import (copy) the Environment Configuration files from the "basic" directory in the newly created directory. When the import completes, the Import Environment Configuration window will then close.

   At this point, the newly created Environment Configuration directory should be identical to the "basic" Environment Configuration.

7. **Remove 'Un-needed" Components of the new Environment Configuration**
   Depending upon the requirements of the new Environment Configuration, it is likely that there may exist DDMs (files) and defined users that need to be dropped from the new configuration.

   To delete unwanted DDMs (files) from the Environment Configuration, the Administrator would use the **Delete DDM** function. This function is accessed from an empty NatQuery desktop by first clicking on **Administer** drop-down menu, then clicking on **Environment Configuration**, then clicking on **DDMs / FDTs**, and then clicking on **Delete DDM**. The Delete DDM function will not only handle the deletion of specific DDMs, it can optionally handle the removal of DDM-related configuration information such as Descriptor Statistic Information, Occurrence Information, File Relationship Information, etc…). Specific information on using the **Delete DDM** function is available by clicking the **Help** button while on this form.

   To delete unwanted defined Users from the new Environment Configuration, the Administrator would use the **Administer User Information** function. This function is accessed from an empty NatQuery desktop by first clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration**, then clicking on **Server Connection Configuration**, and then click on **User Information**. Specific Information on using the Administer User Information function is available by clicking the Help button while on this form.

8. **Complete Administration of new Environment Configuration**
   At the point in time where all extraneous DDMs and defined Users have been removed from the Environment Configuration, the Administrator can begin to provide NatQuery with information pertaining to the new environment configuration.

   Depending upon the nature of the changes required, it may be that all JCL / Script templates may need to be reviewed (for example if a different Batch Natural nucleus will be used).

For detailed instructions on how to proceed, please refer to the section entitled Environment Configuration.

Controlling User Data Access

There are potentially many benefits that an organization can realize by providing End-Users with a Query Tool through which they can directly extract data. While there are many positives to providing this type of data access to End-Users, because this access will make use of production resources, an important consideration is the ability to control this access so that it does not / will not negatively impact production systems.

With NatQuery, there are several methods that will allow an Administrator to regulate and control the impact that End-User requests can have on a production system. These are:

- **Controlling Batch Execution**
  When NatQuery is used against mainframe systems – the simplest way to control Batch execution is to control the Job Class that NatQuery submits into. This can be controlled by "hard-coding" a specific job class into a JCL template (using the Administer JCL / Script function), or it can be controlled by using the High-priority and Low-priority Job Class substitution values (and then allow the user to select High or Low priority – which will dynamically substitute the appropriate value into the Dynamic-Substitution Variable &&JOB-CLASS). For further information on controlling batch access on mainframe, please refer to the section entitled Controlling Batch Execution (Mainframes Servers Only).

  When NatQuery is used against non-mainframe systems – then physical execution of NatQuery tasks will either occur under the control of Remote Execution capabilities (such as RSH, SSH or REXEC), under the control of scheduling software (such as a custom built cron task or through third-party scheduling / submission) or possibly under the control of manual execution. Of these approaches – the use of scheduling software or how the cron task is built can accommodate providing control over Batch Execution.

- **Place Data Access Limits on Users**
  NatQuery allows an administrator to designate Data Access limits on individual users, such that the queries that an individual users runs will automatically stop themselves once the pre-determined number of records has been exceeded.

  For further information on placing Data Access Limits on Users, please refer to the section entitled Place Data Access Limits on Users.

- **Place Statement Restrictions on Users**
  NatQuery allows for users to be restricted in what types of data access statements they may generate. Individual user can be allowed or disallowed to use such statements as READ PHYSICAL or READ BY ISN. Additionally, NatQuery allows the use of FIND statements to either be completely prevented, or otherwise used under tightly designated situations.

  For further information on placing statement restrictions on user, please refer to the section entitled Place Statement Restrictions on Users.

For a discussion relating to how NatQuery can utilize a READ PHYSICAL / READ BY ISN, please refer to the section entitled General Discussion Regarding Use of Read Physical / Read By ISN.

Controlling Batch Execution (Mainframes Servers Only)

Due to the fact that NatQuery data extractions are designed to execute in batch, and given that the NatQuery Administrator has direct control over NatQuery JCL / Script templates and can additionally control the Job Classes which JCL will execute under - one method of controlling End-User data access is to control the Job Classes which will execute these JCL streams.

When a NatQuery Administrator is configuring NatQuery, one of the pieces of required information is the designation of “high priority” and “low priority” Job Classes. This information is supplied to NatQuery through the **Job Priority** tab of the **Administer FTP Server Information** function.

The “high priority” and “low priority” Job Classes are used by NatQuery in the following manner:

- The Administrator is given the ability to set the “high priority” and “low priority” Job Classes to either the same or different values.

- When the Administrator builds the various JCL templates that his / her users will interact with, the Administrator is given the option of “hard coding” a specific Job Class into the template, or the Administrator can make use of the dynamic substitution parameter &&JOB-CLASS.

- When a user submits an extraction request to the server, the End-User is given the ability on the **Send to Server Options** window of requesting that the request be run as “**High Priority Execution**” or “**Low Priority Execution**”. By default, “Low Priority Execution” will be assumed.

- If the Administrator makes use of the dynamic substitution field &&JOB-CLASS in a JCL template (such as the **Production Request Process** template) and the user lets the extract default to “Low Priority Execution”, then when the extraction request is processed through the NatQuery dynamic variable substitution routines, the &&JOB-CLASS variable in the generated JCL will be substituted with the Administratively-provided value for the designated “low priority” Job Class.

  If the Administrator makes use of the dynamic substitution field &&JOB-CLASS in a JCL template (such as the **Production Request Process** template) and the user specifies “High Priority Execution”, then when the extraction request is processed through the NatQuery dynamic variable substitution routines, the &&JOB-CLASS variable in the generated JCL will be substituted with the Administratively-provided value for the designated “high priority” Job Class.

  If the Administrator does not make use of the dynamic substitution field &&JOB-CLASS in the **Production Request Process** template, then this JCL template must have a “hard-coded” Job Class in order to allow it to execute. In this case, it makes no difference whether the End-User selects “High Priority Execution” or “Low

Priority Execution" as the generated JCL will contain the "hard-coded" administratively provided value.

Depending upon how Job Classes are utilized at a customer site, the Administrator can exercise a great deal of control over when NatQuery-generated data extracts execute, as well as potentially controlling how long extracts execute (both of which are controlled by the attributes of a specific initiator).

Place Data Access Limits on Users

As a method of controlling any potential negative impact to production systems, the Administrator is enabled to place specific data access limits on each user. In the current version of NatQuery, the Administrator can place the following Data Access Limits on each user defined to NatQuery:

- An upper limit can be placed on the total number of records handled (by the outermost I/O loop) on generated extracts. This limit is implemented with a generated Natural program as an incremented record counter – if the counter exceeds the specified limit, the generated Natural program will terminate with an error.

The Administrator can selectively place the above limitations on individual users through the **Data Access Limits** tab of the **Administer User Information** function.

When upgrading from NatQuery version 2.4.1 to version 2.4.2 (or higher), NatQuery will assume that all existing users will be allowed to perform a Read Physical, and will be limited to only extracting 20,000 records.

Place Statement Restrictions on Users

As another method of controlling any potential negative impact to production systems, the Administrator may either restrict or allow the use of specific Natural I/O statements. Such statement restrictions would be in addition to the Data Access Limits described above.

In current versions of NatQuery, the Administrator can:

- Allow or disallow a user to be able to use the **READ PHYSICAL** statement.

  In specific situations, a **READ PHYSICAL** may be arguably the best statement to use when it is absolutely necessary to access every record in a file, Data Warehousing for example. Or, in situations where records must be selected against fields which are not Descriptors and / or part of usable Super-Descriptors.

  Whether or not a given user may generate a **READ PHYSICAL** statement as a result of a query is determined through the **Administer User** function, accessed from an empty NatQuery desktop by first clicking **Administer / Environment Configuration / Server Connection Configuration / User Information**.

- Allow or disallow a user to be able to use the **READ BY ISN**

  In specific situations, and in order to insure that an extraction process against a given file will in fact only extract a given unique record only once – the **READ BY ISN** has its place.

  Whether or not a given user may generate a **READ BY ISN** statement as a result of a query is determined through the **Administer User** function, accessed from an empty NatQuery desktop by first clicking **Administer / Environment Configuration / Server Connection Configuration / User Information**.

- Allow or disallow when a **FIND** statement can be utilized.

  Due to the adverse effect of the way a FIND statement results in the internal creation of an ISN list, the FIND statement is typically best suited for use when it is known that only a small number of records will be handled (therefore resulting in a small ISN list).

  Whether or not a given user may generate a **FIND** statement as a result of a query is determined through the **Administer I/O Parameters** function, accessed from an empty NatQuery desktop by first clicking **Administer / Environment Configuration / I/O Parameters** function.

General Discussion Regarding Use of Read Physical / Read By ISN

While there are many situations where the most optimal access method to resolve a given query will be through the use of the Natural READ PHYSICAL statement, the use of this statement in Natural carries with it implications of possible negative impact on Production Systems.

Similarly, the use of a READ BY ISN has its place (to insure unique records are handled only once), however this statement also has potential negative implications to the performance of Production Systems.

In considering the possible use of a READ PHSYICAL; it should be noted that the negative potential impact of such access may be greatly reduced through the use of ADABAS facilities such as the Pre-Fetch / Multifetch feature. NatQuery can automatically support the use of Multifetch, and can generate the appropriate statements so as to enable this feature of ADABAS. For more information on using this feature of NatQuery, please see the section of this manual entitled Support for ADABAS Multifetch.

NatQuery's ability to determine an optimal access path to resolve a query is directly related to the Selection Criteria that the user specifies. The appropriateness of a READ PHYSICAL or a READ BY ISN may therefore directly result from a query that utilizes a Selection Logic statement of "Read All Records From Filename". Alternatively, it may result from typical Selection Logic statements which do not refer to Descriptors or as usable component fields of available Super-Descriptors.

In a situation where NatQuery detects that the entire file should be read in order to resolve a query, the following rules will apply:

- The User must be allowed to use READ PHYSICAL or READ BY ISN

  When NatQuery determines that a READ PHYSICAL or READ BY ISN is an appropriate statement to use to resolve a query, NatQuery will examine the administratively created attributes of the user's profile.

  If the End-User is not allowed to generate a READ PHYSICAL or READ BY ISN, the user will be prevented from submitting the query for execution and will be provided with a Message box that explains the situation as well as providing instructions as to how to correct it (I.E., refine existing Selection Logic)

- If the End-User is authorized to generate a READ PHYSICAL and a READ BY ISN, then NatQuery will present the user with a message that will discuss the merits of one over the other. The user may then select READ PHYSICAL, READ BY ISN or may Cancel the generation process.

  If Cancel is not specified, then NatQuery will allow the query to be submitted.

- If the End-User IS allowed to generate a READ PHYSICAL but not a READ BY ISN, or vice-versa, and they directly use a Selection Logic statement of "Read All Records From Filename", then NatQuery will proceed to generate the query with whichever statement the user has been given permission to use, and NatQuery will then allow it to be submitted.

Suppression Handling Information

The most important thing to know about setting suppression options is that the use of suppression on a descriptor or super-descriptor directly affects whether or not records are returned from ADABAS. The biggest problem with this is that descriptors and/or super-descriptors that are marked as null-suppressed (which is a common practice) may not return all of the records in a given file, because any records not having a value for a given descriptor / super-descriptor will be suppressed (that is, not returned by ADABAS).

Suppression information is one of the factors that helps NatQuery determine whether or not to use a Descriptor or Super-Descriptor when executing a query. If suppression handling information is not present, the end-user may be prompted to provide information to NatQuery to determine if a particular super-descriptor can be considered for use as a data access path into a file.

Because this suppression information is used by NatQuery to determine how to generate an optimal Natural program, the Administrator should pay special attention to this field. If a Descriptor is marked as affected by suppression, NatQuery will not consider that Descriptor as an access path into a file, unless the descriptor is in some way referenced by selection logic.

Other aspects of Suppression handling:

- How NatQuery Determines Suppression Handling Settings,
- Manually Modifying Suppression Handling Information, and
- Use of Suppression Handling Information in Building Queries.

How NatQuery Determines Suppression Handling Settings

NatQuery will only attempt to set suppression handling information automatically during an import of descriptor statistics. When descriptor statistics are imported into NatQuery, it attempts to calculate the total number of records in the file from the descriptor and super-descriptor information. If NatQuery cannot do this, the Administrator will be prompted to enter a total for NatQuery to use for this file. If the Administrator does not know the total number of records in a given file, they can run an ADAREP report to get this information.

It is recommended that the Administrator let NatQuery (on import of Descriptor Statistics) determine which Descriptor / Super-Descriptors are affected by suppression, but the suppression option for descriptors can be set after import. NatQuery determines which descriptors are affected by suppression by first looking at the suppression field for the descriptor in the DDM that the descriptor is from. If the descriptor is not null-suppressed, NatQuery considers it to be unaffected by suppression. If it is null-suppressed, NatQuery takes the total number of records associated with the descriptor and compares that number to the total number of records in the file (which was determined earlier). If these two numbers are equal, the descriptor is considered to be unaffected by suppression. The reason for this is that even though the descriptor is null-suppressed, all fields have a value for the descriptor, so there are no null values for the descriptor that will cause records to be suppressed.

NatQuery treats components of super-descriptors in the same way as descriptors. The only difference is that once NatQuery has calculated the suppression handling settings for all of the components of a super-descriptor, it also makes a determination of whether or not the super-descriptor is affected by suppression. If all of the components of a super-descriptor are not affected by suppression, the super-descriptor is not affected by suppression. Otherwise, if a component is affected by suppression, the super-descriptor is affected as well.

Manually Modifying Suppression Handling Information

Suppression information can be modified manually, but it is best to let NatQuery handle suppression information automatically. Descriptors can be changed directly just by selecting the descriptor (in the grid on the **Administer Descriptor Statistics** window), changing the value of the **Suppress** combo-box, and clicking the **Apply** button. Suppression information for a Super-Descriptor cannot be modified directly, each of its components has suppression information that can be set, which NatQuery then uses in order to set the suppression information for the Super-Descriptor.

Because suppression handling information can make the difference between which descriptor or super-descriptor is used by NatQuery as a data access path, and because it can impact whether or not all desired records are actually extracted, caution should be used when making manual changes to these settings.

Use of Suppression Handling Information in Building Queries

When NatQuery generates a query program, either to send to the server or just for a listing, it looks for the best data access path into the file(s) associated with the query's Selection Logic. NatQuery does this by looking at all the descriptors and super-descriptors in a file, and determining which ones can be used. Only those descriptors / super-descriptors that have been referenced can be consider for use, and of those, only the ones not affected by suppression will be considered. This is the point where suppression handling information plays an important role.

If NatQuery cannot determine from the DDM that a field will not be affected by suppression, then NatQuery will look up the suppression handling setting for that descriptor / super-descriptor, and handle the field appropriately. If no suppression information has been supplied for a given file, then the end-user may be prompted to tell NatQuery whether or not certain super-descriptors may be used, if NatQuery cannot determine this on its own.

It is recommended, for the sake of your end-users, that any information that shields them from any decision making process regarding super-descriptors is applied to the appropriate DDM(s).

NatQuery Support for Security

NatQuery supports any and all existing levels of security that may exist in a customer's site.

The first level of security is at the server level itself. In order to connect to the Natural Server platform, a user of NatQuery must have a valid FTP User-ID and they must know the valid password associated with this user.

A second level of security is at the Natural Security or ADABAS Security, with Natural Security being an add-on product, and ADABAS security being available as part of ADABAS option ADASCR. For information on the use of Natural and ADABAS Security, please refer to the section entitled Natural Security or the section entitled ADABAS Security.

A third level of security is any existing network-level security that may exist in a Windows environment into which NatQuery may be deployed. When NatQuery users are provided with a shared network directory for use as an Environment Configuration Path, Windows can apply restrictions such that only specific users have access to this directory. Further information on Network Security can be found in the section entitled Network Security.

A final level of security is NatQuery's internal handling of a key environment configuration file (VALIDENV.CFG). This file contains information on dates and times of DDMs and configuration files at the time these files were last verified by an Administrator through the Verify Environment Configuration function. Only Administrator versions of NatQuery can alter this file, and then only indirectly, through the **Verify Configuration** function. This means that any end-user who alters DDM files or configuration files directly will not be able to update this key configuration file, and will thus not be able to use NatQuery.

Natural Security

Batch Natural environments protected by the Software AG product Natural Security require additional statements in Job Control Language (JCL) in order to allow secured access. Typically, the statements required will be a Natural Library name, a valid User-ID and a valid Password that is associated with the given User-ID.

In order to allow NatQuery JCL templates to work with Natural Security, the NatQuery Administrator can utilize one of two approaches:

- NatQuery can be instructed to prompt for Natural Security User-Ids and Passwords at the time a request is submitted for execution, with the captured User-ID and Password being dynamically substituted into NatQuery JCL / Script templates. Using this approach, NatQuery will completely embrace Natural Security in the tightest possible manner.

- The Administrator can "hard code" appropriate User-IDs and Passwords directly into the respective NatQuery JCL templates. While this approach will allow access through Natural Security, as this method essentially bypasses the intent of secured access, NatWorks does not recommend this approach.

Further topics of discussion concerning Natural Security:

- NatQuery Processing and Natural Security, and

- Configuring NatQuery to work with Natural Security

NatQuery Processing and Natural Security

NatQuery support for Natural Security is "turned on" by the NatQuery Administrator through the **Natural Security Installed** checkbox located on the **Administer Natural Server Information** window.

Once the **Natural Security Installed** checkbox is selected, this enables three separate processes within NatQuery:

- Two Natural Security-specific dynamic substitutions variables will be made available for use within the Administer FTP JCL / Script function. These variables are named &&NATSEC-USER-ID and &&NATSEC-USER-PWD.

- Each time an End-User requests submission of an extraction request to the server, NatQuery will prompt the user with a **Natural Security User ID / Password** window that is designed to capture a Natural Security User-ID and Password. The captured User-ID and Password ARE NOT retained (stored) by NatQuery past the users open session - NatQuery will however prompt for validation of the values every time a user submits a request for execution during the open session.

- Within the process that handles the substitution of dynamic variables for JCL / Script templates prior to submission, NatQuery will search for the Natural Security-related dynamic variables and will replace all references found with the values captured from the **Natural Security User ID / Password** window.

Configuring NatQuery to work with Natural Security

To configure NatQuery to fully support Natural Security, perform the following steps:

1. **Inform NatQuery that Natural Security is Installed**
   NatQuery is informed that Natural Security is installed through the **Administer Natural Server Information** function. While on an empty NatQuery desktop, this function is available by first clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration**, and then clicking on **Natural Server Information**. These actions will invoke the **Administer Natural Server Information** function.

   On the **Administer Natural Server Information** window, there is a checkbox labeled as **Natural Security Installed**. To enable NatQuery to automatically handle Natural Security, insure that the checkbox labeled **Natural Security Installed** is selected (checked).

   Once the Natural Security Installed checkbox is selected, you can now close the **Administer Natural Server Information** window by clicking the **OK** button. As the window closes, NatQuery will sense that core configuration information has changed and NatQuery will prompt with a message asking if the changed information should be verified. At this time it is suggested that you click **No** to this prompt.

2. **Modify JCL / Script Templates**
   With the **Natural Security Installed** checkbox option selected, NatQuery will then make available two NatQuery Security-specific dynamic substitution variables for use in JCL templates. These dynamic variable are named &&NATSEC-USER-ID and &&NATSEC-USER-PWD. These variables can then be placed into NatQuery JCL / Script templates at the appropriate job steps that invoke batch Natural.

   In most cases, Natural Security will require a Natural Library (which the user will be logging onto) as well as values for User-ID and User Password. For the Natural Library value, the Administrator can either utilize the dynamic variable &&USER-NATURAL-LIBRARY (this will be the value of **Natural Library** as specified through the **Administer Natural Server Information** function), or can optionally hard-code the Natural Library to a specific value. The values for User-ID and User-Password can either be replaced with the Natural Security-specific dynamic substitution variables &&NATSEC-USER-ID and &&NATSEC-USER-PWD, or these values can be hard-coded.

   When installing Natural Security related dynamic-variables in NatQuery JCL Script templates, the Administrator should typically only need to handle the **Production Request Process** and the **Production Special Process** templates.

   When using "Just" FTP, the Administrator would NOT install Natural Security variables into the Production Server Process template, as this template will require that these Natural Security values be hard-coded.

When installing Natural Security-related variables into NatQuery JCL / Script templates, the Administrator may find it advisable to make use of the Natural Terminal Command of “%*”. This command, when issued in a batch-Natural input stream, will cause NatQuery to suppress the printing of the next record input (thus hiding the Natural Security string from being printed).

3. **Verify Environment Configuration**
   Once the Administrator completes the changes to the two NatQuery JCL / Script templates, the Environment Configuration should then be verified. NatQuery will automatically prompt for this when the Administer JCL / Script window is closed, or this can be done subsequently by invoking the Verify Environment Configuration function.

ADABAS Security

Any files that have been assigned an ADABAS password require an extra statement in any generated Natural code that will run on the server. NatQuery provides support for this, and can be configured to prompt users for an ADABAS password for a given file (or files). Unlike Natural Security or FTP, ADABAS security does not require a username, only a password.

To configure NatQuery to work with ADABAS Security, please refer to the section entitled Configuring NatQuery to use ADABAS Security.

Configuring NatQuery to use ADABAS Security

The ADABAS Security option is found on the **Administer Descriptor Statistics** window.

Checking the **ADABAS Password-Protected** checkbox turns on ADABAS Security for the selected file. This will have two effects in NatQuery. The first is that any time a file that requires an ADABAS password is included in a query, NatQuery will prompt the user for the appropriate ADABAS password.

The second is that after prompting for this password, NatQuery will generate a PASSW statement into the Natural program to be run on the server. This is a simple statement, generated before the file access requiring the password, and follows this format:

```
PASSW=’password’
```

Where ‘password’ is the password entered earlier by the user.

Network Security

Due to the nature of how NatQuery user's must interact with the shared network directory a network directory can be secured to only support defined network users by any existing network security.

Authorized users of NatQuery must be defined to have Read access to files in the specified Environment Configuration Path, and additionally must have Write permissions as well.

Predict

In most organizations that have ADABAS, the Software AG product Predict is utilized to help manage various aspects of the application environment. While Predict has many uses and abilities, one of the more common uses of Predict is to generate Natural Data Definition Modules (DDMs). NatQuery can obtain all element level documentation from Predict, which can be useful in helping end-users and administrators understand the purpose and layout of a given field.

Please refer to the following topics related to the use of Predict:

- Obtaining Documentation from Predict, and

- DDM Generation with Predict.

Obtaining Documentation from Predict

In retrieving information from Predict, NatQuery makes use of a JCL / Script template called **Production Predict Process**, with this template being specific to the execution of a Predict Display utility in batch.

Once this JCL / Script template is correctly created, NatQuery provides the **Request Predict Information** function that allows for the generation of requests that will extract appropriate field-level information from Predict. These requests are similar to a typical NatQuery data extract request in that this request will appear in the **Check Server** window. Once this request is "**DONE**", NatQuery will allow for the downloading of the Predict documentation, which will then be available for display - both to end-users and administrators - from certain points in NatQuery.

Administrators will be able to modify this data as they see fit, while end-users will only be able to view this information.

To create the **Production Predict Process** template, the Administrator should refer to the section entitled JCL / Script Templates.

To submit Predict requests, the Administrator would use the **Request Predict Information** function. This function would be accessed on an empty NatQuery desktop by clicking on **Administer / Environment Configuration / Predict / Request Predict Information**.

As an alternative to using the above NatQuery functionality to submit a Predict Request, this information can be imported into NatQuery using the **Import Predict Information** function because NatQuery utilizes the Predict Direct Command of "DIS ELE * *Filename*". This function would be accessed from an empty NatQuery desktop by first clicking on **Administer / Environment Configuration / Predict / Import Predict Information** window.

As another alternative to using NatQuery functionality to submit a Predict Request, it should be noted that Administrator versions of NatQuery allow for field information to be manually entered against fields such that users can subsequently use this information in the same way as if this information had originated from Predict. This would be accomplished by an Administrator by creating a query against a file, and while on the **Select Fields from *filename*** window, the administrator right-clicks on a field name in the left-hand pane (labeled **Possible Fields to Select**), and then on the resulting **Predict Information for *fieldname*** window – enters and needed descriptive text and then clicks the **OK** button.

DDM Generation with Predict

When Predict is used to generate DDMs, there are several options available within Predict that control what information is physically placed into a generated DDM. When DDMs are to be used with NatQuery, there are two-generation options that either must be utilized or should be utilized in order to achieve the highest level of integration to NatQuery.

Two "rules" should be followed when providing NatQuery with Predict generated DDMs:

- DDMs Must Contain the Composition of Super and Sub Descriptors, and
- Predict Handling of Occurrence Information.

DDMs Must Contain the Composition of Super and Sub Descriptors

In order for NatQuery to understand the composition of Super-Descriptors and Sub-Descriptors that may be contained in a DDM, it is a requirement that all DDMs be generated so that Super-Descriptors and Sub-Descriptors have the field components that comprise these fields be documented in the DDM. An example of this type of documentation would be:

```
TYL  DB  NAME                              F LENG  S D REMARKS
---  --  --------------------------------  - ----  - - ----------------------

  1  S1  DEPARTMENT                        A    4    S
         HD=SECTION
*        -------- SOURCE FIELD(S) -------
*        DEPT(1-4)
  1  S2  DEPT-PERSON                       A   26    S
*        -------- SOURCE FIELD(S) -------
*        DEPT(1-6)
*        NAME(1-20)
```

In the above example, the field DEPARTMENT is a Sub-Descriptor that is comprised of the first four bytes of the field DEPT. In a similar fashion, the field DEPT-PERSON is a Super-Descriptor that is comprised of the first 6 bytes of the field DEPT, and the first 20 bytes of the field NAME.

To enable the generation of this component information from Predict, when a DDM is generated from Predict the DDM should be generated so that the generation option **General Comments** is set to “Y”.

Predict Handling of Occurrence Information

In order to assist the Administrator with the creation of **Occurrence Information** for any Multi-Valued fields (MUs) or Periodic-Groups (PEs) that may be present in a given DDM, it is a recommendation that DDMs be generated so that any documented occurrence information is included as comments in a given DDM.

If this information exists in Predict (it may not), then this information can be optionally output into a Predict DDM. An example of this information would be:

```
TYL  DB  NAME                              F LENG  S D REMARKS
---  --  --------------------------------  - ----  - - ----------------------

M 2  AI  ADDRESS-LINE                      A   20  N   Max. occurrences 3
         HD=ADDRESS
```

With this information available in a DDM, NatQuery can then utilize this information to assist the Administrator in the task of entering Occurrence Information when the Administrator builds an Environment Configuration.

To enable the generation of any documented occurrence information from Predict, when a DDM is generated from Predict the DDM should be generated so that the generation option **Line Comments** is set to “O”.

While not being an absolute requirement that DDMs contain occurrence information, it is strongly suggested that the **Line Comments** option be used when generating DDM from Predict that are destined to be used by NatQuery. Doing so will greatly speed the administrative process of defining Occurrence Information to NatQuery for a given file (DDM).

Support for ADABAS Multifetch

Multifetch is a feature of ADABAS that can greatly reduce mainframe processing time while running certain types of commands. Currently NatQuery only supports Multifetch for READ PHYSICAL statements and Histograms as the benefits of using Multifetch with these statements can be tremendous.

NatQuery fully supports Multifetch through the automatic generation of ADARUN parameters that will invoke the Multifetch feature. This includes proper generation of PREFSBL, PREFTBL, PREFXFIL and PREFXCMD option statements.

Multifetch handling is turned on through **Administer Natural Server Information** window in NatQuery. To turn on this feature, make sure the **Multifetch Enabled** option (located in the **Multifetch Options** frame) is checked. The Administrator also has control over the PREFSBL and PREFTBL options, which are the **Buffer Length per Call** option, and the **Total Size of Buffer** option, respectively. Defaults are provided for all these values, but the Administrator should check to make sure that these values are appropriate for their site.

In order to utilize Multifetch, it is typically necessary to allow a given JCL / Script the ability to pass statements to ADARUN. Example of such statements would include:

```
ADARUN PROG=USER,PREFETCH=YES
ADARUN PREFSBL=32767,PREFTBL=327670.
```

For further information on the Multifetch feature, please refer to the ADABAS Operations Manual and/or the ADABAS Command Reference Manual

Configure NatQuery to use FTPS against a RDBMS Target

This section describes the steps needed to configure NatQuery to use a Secure FTP connection against the platform upon which a target RDBMS resides. Depending upon how your organization uses NatQuery / NatCDC this section may not be applicable; it will only be applicable if you are using NatQuery / NatCDC to extract data from ADABAS and load this data into a RDBMS residing on a completely separate platform.

To configure Secure FTP against a remote RDBMS platform, perform the following steps:

1. **Start NatQuery**
   Please start NatQuery if it is not already started. If already started, insure that NatQuery is open with an empty NatQuery desktop (I.E. no Query windows or other windows open).

2. **Invoke the Administer Server Connection Information Window**
   On an empty NatQuery desktop, click **Administer** > **Environment Configuration** > **RDBMS Target Configuration** > ***RDBMSName*** (where the "***RDBMSName***" is either "SQL Server", "MySQL" or "Oracle". This action will invoke appropriate **RDBMS Target Configuration –** ***RDBMSName*** **- General Defaults** window.

   When this window appears, the first Tab entitled ***RDBMSName*** **Command Options** (where ***RDBMSName*** is either "SQL Server", "MySQL".

   Click on the **Execution Configuration** Tab to continue.

3. **Configure FTP – Execution Configuration Tab**
   The Execution Configuration Tab provides for the capture of FTP related information.

   a. **Server Type**
      Set the Server Type to the value that represents the type of platform upon which the Target RDBMS resides. Possible values are "Windows" or "UNIX/Linux".

   b. **Transport Mode**
      Set the Transport Mode to be the desired communication Mode. Possible values are "PC Network/Filecopy", "FTP" and "none".

      If Secured FTP communication is desired, then this value should be set to "FTP".

   c. **Server Name**
      Set the Server Name to be the Universal Resource Locator (URL), Name or IP Address of the platform upon which the RDBMS Server resides.

   d. **Remote Execution Enabled**
      This checkbox instructs NatQuery as to whether or not NatQuery will attempt to execute the NatQuery-generated processing scripts once they are placed into the target machine.

Further discussion on configuring Remote Execution is described in the NatQuery Installation and Operations Manual.

e. **FTP Information Frame**

i. **Encryption**
This should be set to the type of FTP connection that is desired. Options are “None (Normal FTP)”, “Implicit FTPS” and “Explicit FTPS”; select the type of FTP communication that matches the setting of the FTP Server on the remote RDBMS platform.

ii. **Port**
This should be set to the value of the Port on the remote FTP Server on the RDBMS target that will handle the FTP Connection.

For normal FTP, this is usually “21” and in most cases will be “990” by default when FTPS Implicit or Explicit connections are used.

Set the **Port** setting to the correct value.

iii. **Passive FTP**
If checked, this checkbox will enable Passive (PASV) FTP communication, if left unchecked Passive FTP communication will be disabled.

Usually, **Passive FTP** should be checked.

iv. Create FTP Logfile
This checkbox controls whether or not a Log File of FTP Operations is created when a user performs a FTP operation. This Log File can be useful when debugging connections issues, but should be disabled (unchecked) when FTP operations are working properly because a user’s password **IS** recorded in the Log File.

When checked, a log file with the name of:

*userid*_FTP_RDBMS_TRACE.Log

This file is created in the path specified by the NatQuery Environment Path, where “*userid*” is replaced with the User ID of the NatQuery user.

f. **FTP Transport Frame**
The FTP Transport Frame contains a single field; **Execution Directory.**

The path value placed into this text field is relative to the Target RDBMS platform, and represents the directory that automated FTP will make a Change Directory to, and will additionally be used within the NatQuery-generated script so that execution of Load processes may function correctly.

Set this path to the relative path on the target RDBMS platform where FTPed files will be placed (scripts, parameter files, and data) for subsequent loading into the target RDBMS.

g. **Complete RDBMS Target FTP Configuration**
With the above steps completed, Secure FTP against the target RDBMS platform should now be properly configured.

Click the **OK** button to close the **Target Configuration –** ***RDBMSName*** **- General Defaults** window.

4. **Handle Secure FTP Certificate(s)**
With the above steps complete, you can proceed to handle the Secure FTP Certificate(s) needed to complete the Secure FTP configuration against the target RDBMS platform; these are outlined in the section of this manual entitled **Handling Secure FTP Certificates**.

Handling Secure FTP Certificates

When enabling Secure FTP connections, a client machine needs to be given the appropriate Certificate issued by a Certificate Authority (CA) for the platform being accessed. This Certificate is then stored internally in the Windows Registry where it is subsequently automatically accessed when Secure FTP Operations are executed.

If the appropriate Certificates are already installed into the Security Store on the Client Machine then this section may be bypassed.

In the current version of NatQuery, NatQuery itself does not provide any mechanism to Import a Certificate into the Windows Registry as this ability is already inherent in a Windows environment through a function available with Microsoft Internet Explorer and other browsers.

As Internet Explorer (IE) is typically available in every Windows installation, these instructions utilize IE as the mechanism to Import required Certificates.

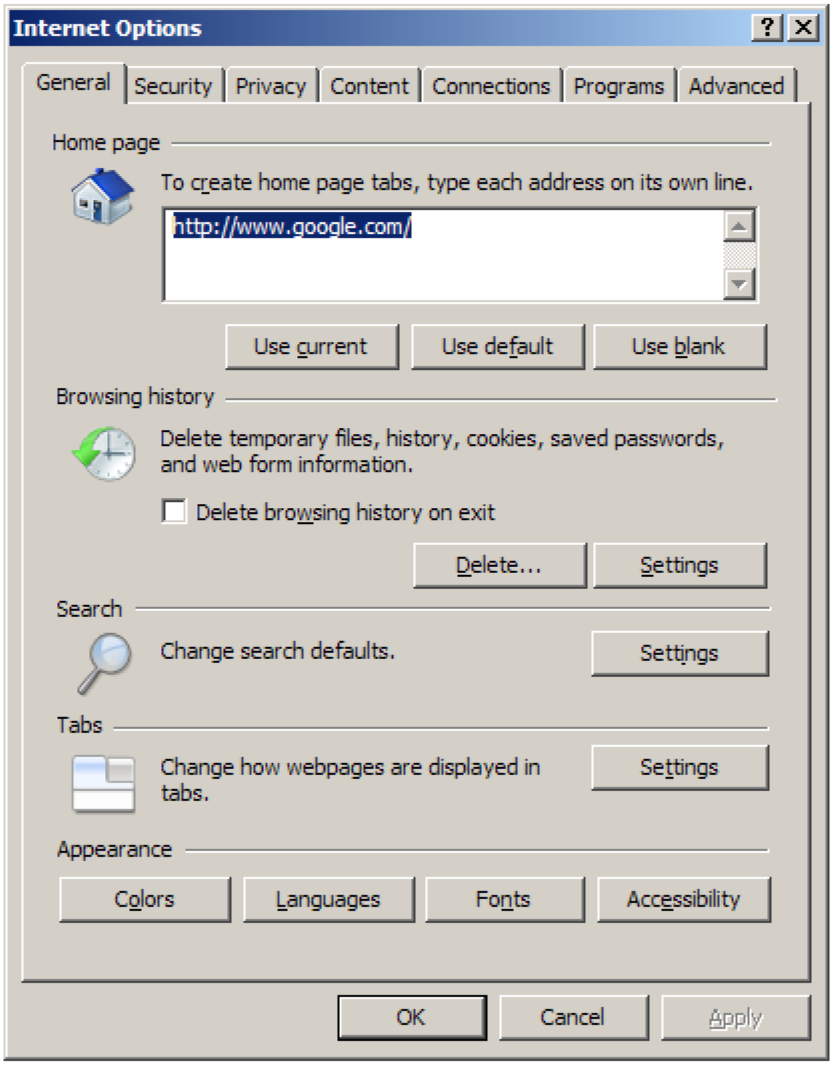
NOTE: The following process assumes that you have computer access to a digital Certificate created by a Certificate Authority that corresponds to the target FTP platform. If you do not have this Certificate – you will not be able to complete the following steps successfully. Please insure you have path access to the Certificate from the computer you are presently working on.

To import a Certificate, perform the following steps:

1. **Start Microsoft Internet Explorer**

   Instructions continue on subsequent pages.

2. **Invoke Internet Options**
   On the IE toolbar will be an item called **Tools**. Clicking on **Tools** will invoke a menu where you will find an **Internet Options** item. Clicking the **Internet Options** item, which will invoke a window similar to the following (the window for IE 8 is shown below):



To create home page tabs, type each address on its own line.

http://www.google.com/

Delete temporary files, history, cookies, saved passwords, and web form information.

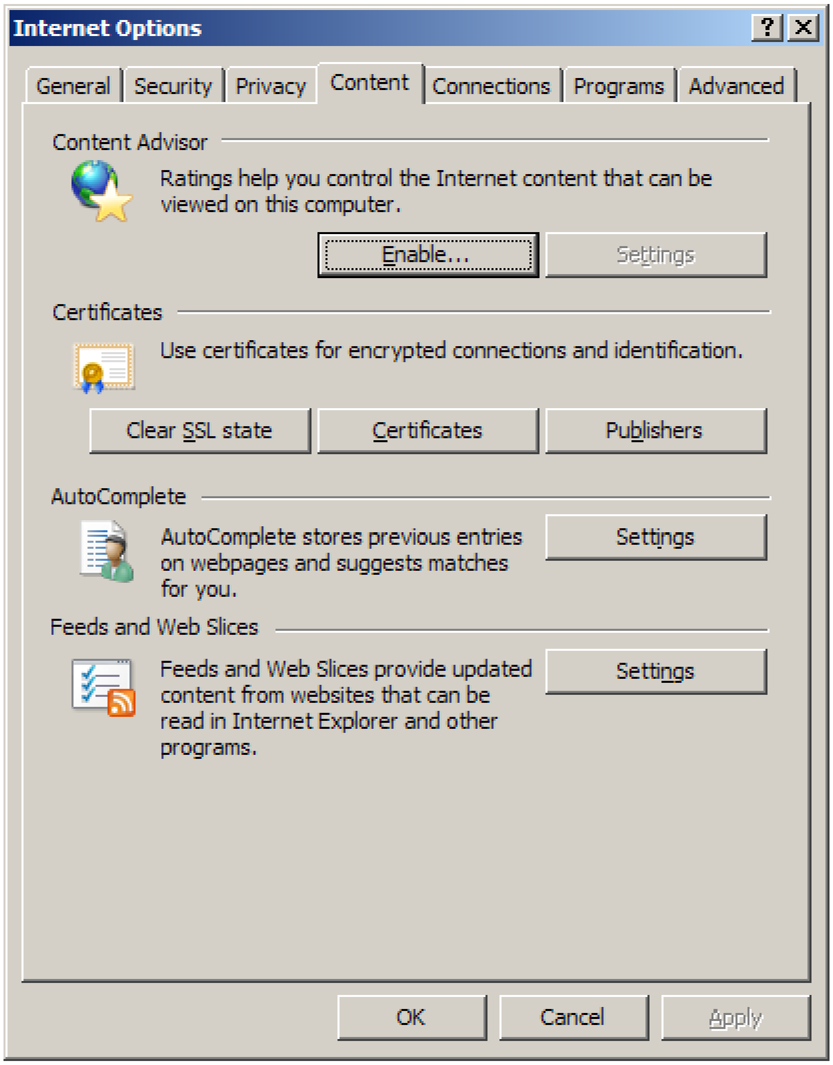
☐ Delete browsing history on exit

Change search defaults.

Change how webpages are displayed in tabs.

To continue, click on the Tab entitled **Content**.

3. **Internet Explorer, Content Tab**
Clicking the **Content** Tab as described above will invoke the **Content** Tab, which will look similar to the following:



Ratings help you control the Internet content that can be viewed on this computer.

Use certificates for encrypted connections and identification.

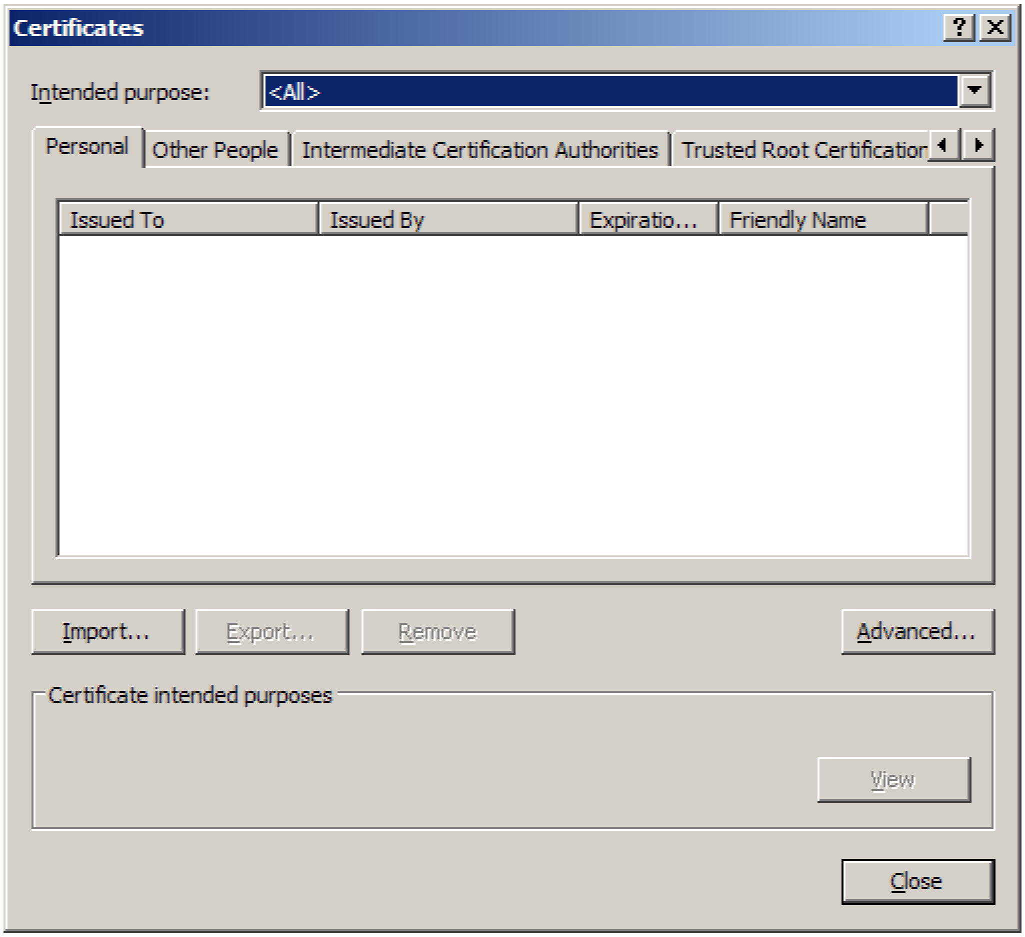
AutoComplete stores previous entries on webpages and suggests matches for you.

Feeds and Web Slices provide updated content from websites that can be read in Internet Explorer and other programs.

To continue, click the **Certificates** button located in the middle of the **Content** Tab. This will invoke a **Certificates** window.

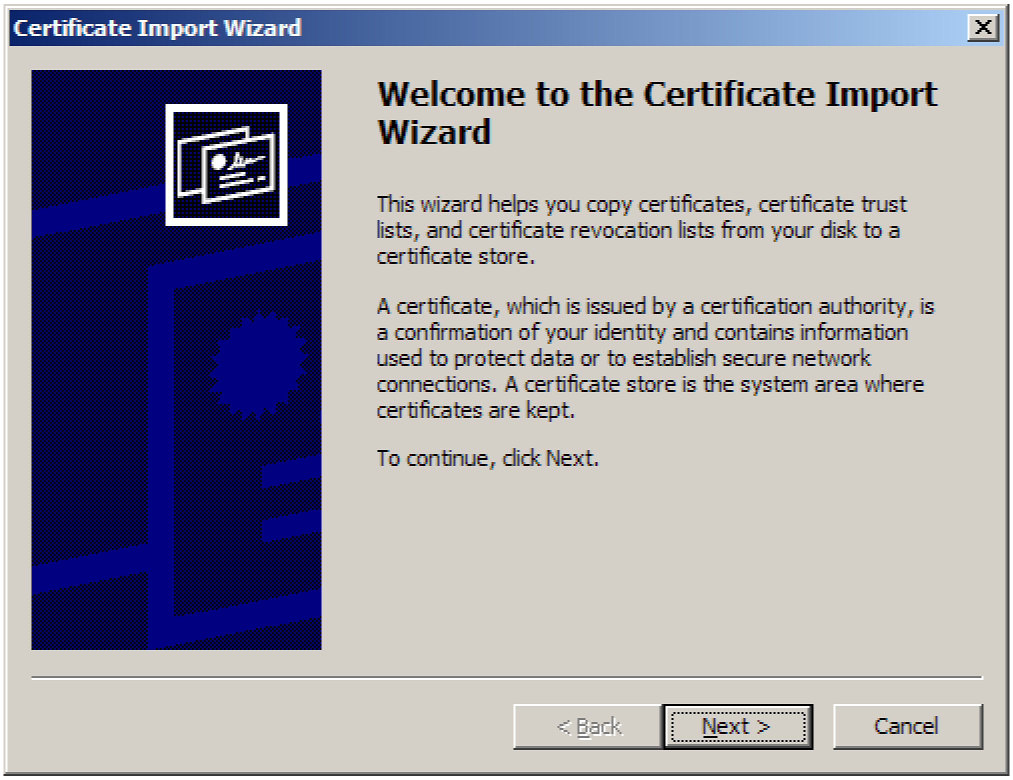
4. Internet Explorer – Certificates Windows

Clicking the Certificates button described above will invoke a window similar to the following:



To continue, click the **Import** button on the **Certificates** window.

5. **Internet Explorer – Certificate Import Wizard – Step 1**
   Clicking the **Import** button as described above will invoke the first window of the **Certificate Import Wizard**, which will look similar to the following:



Certificate Import Wizard

Welcome to the Certificate Import Wizard

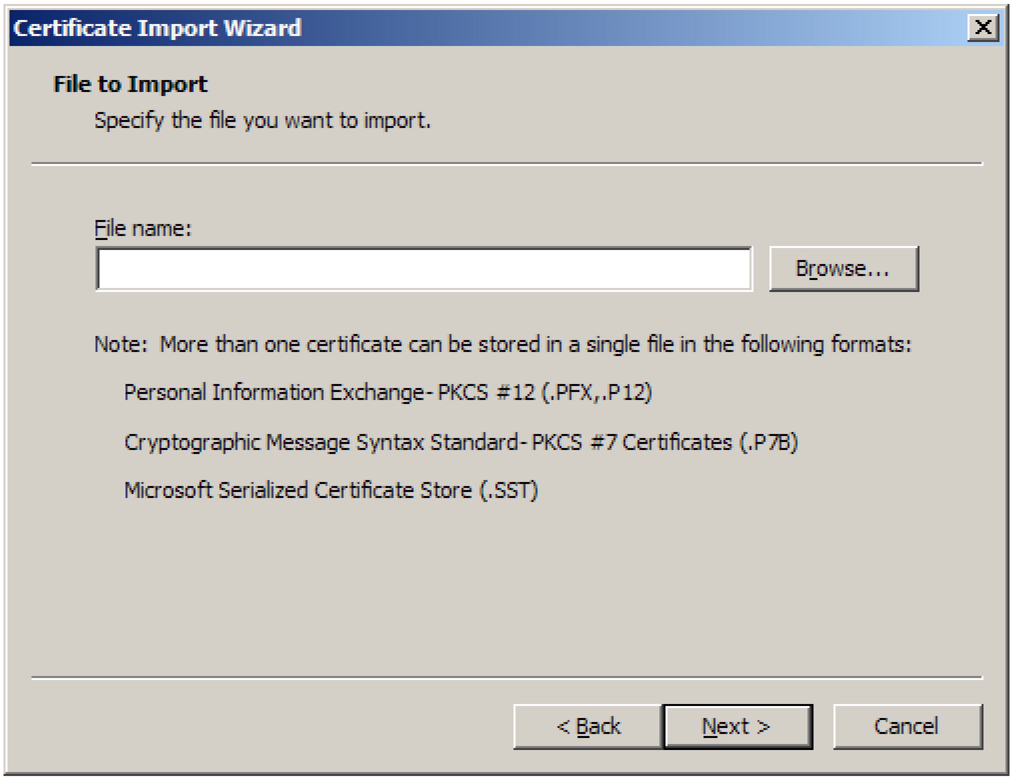
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

To continue, click the **Next** button.

6. **Internet Explorer – Certificate Import Wizard – Step 2**
Clicking the **Next** button as described above will bring up the next screen of the **Certificate Import Wizard**, which will look similar to the following:



Certificate Import Wizard

File to Import
Specify the file you want to import.

File name:

Note: More than one certificate can be stored in a single file in the following formats:

Personal Information Exchange- PKCS #12 (.PFX,.P12)

Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialized Certificate Store (.SST)

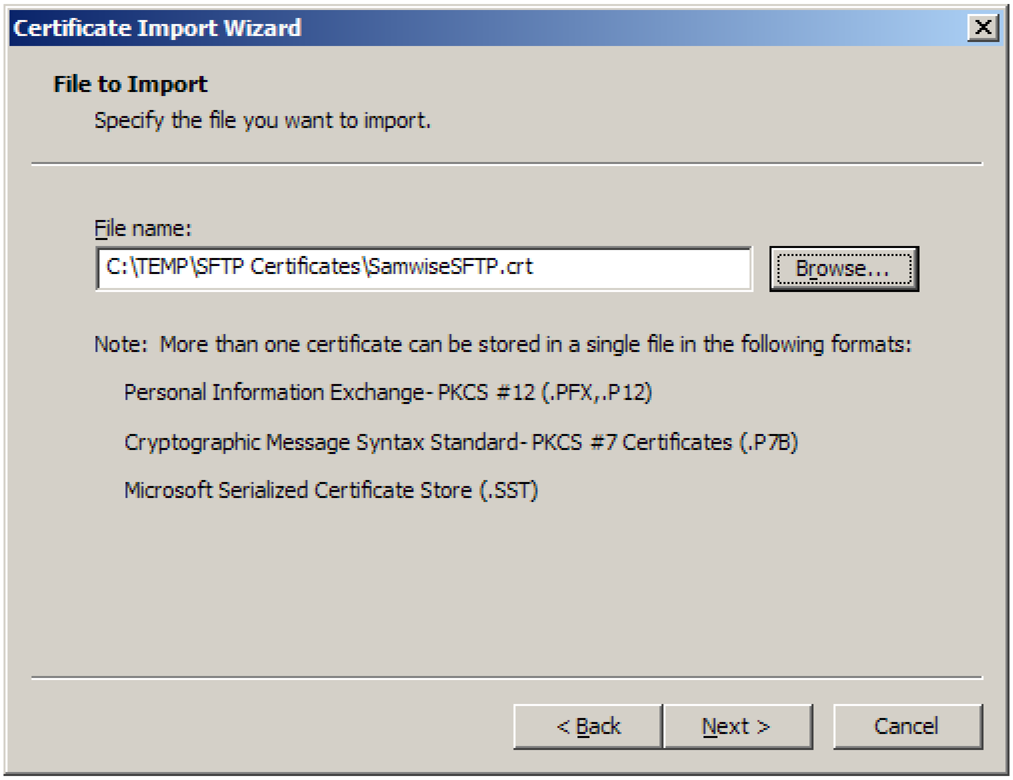
To continue, click the **Browse** button.

7. **Internet Explorer – Certificate Import Wizard – Step 3**
Clicking the **Browse** button as described above will invoke a typical Windows **Open** window that will allow you to navigate to the path where the Certificate that was provided to you has been temporarily stored.

Navigate to the appropriate directory where the Certificate that was given to you has been saved, left-click on it to select / highlight this file.

To continue, then click the **Open** button.

8. **Internet Explorer – Certificate Import Wizard – Step 4**
Performing the above action will return the user to the window seen in #6 above, with the appropriate Certificate file selected. This will now look similar to the following image:


Certificate Import Wizard
File to Import
Specify the file you want to import.
Browse...
< Back
Next >
Cancel

File name: C:\TEMP\SFTP Certificates\SamwiseSFTP.crt

Note: More than one certificate can be stored in a single file in the following formats:

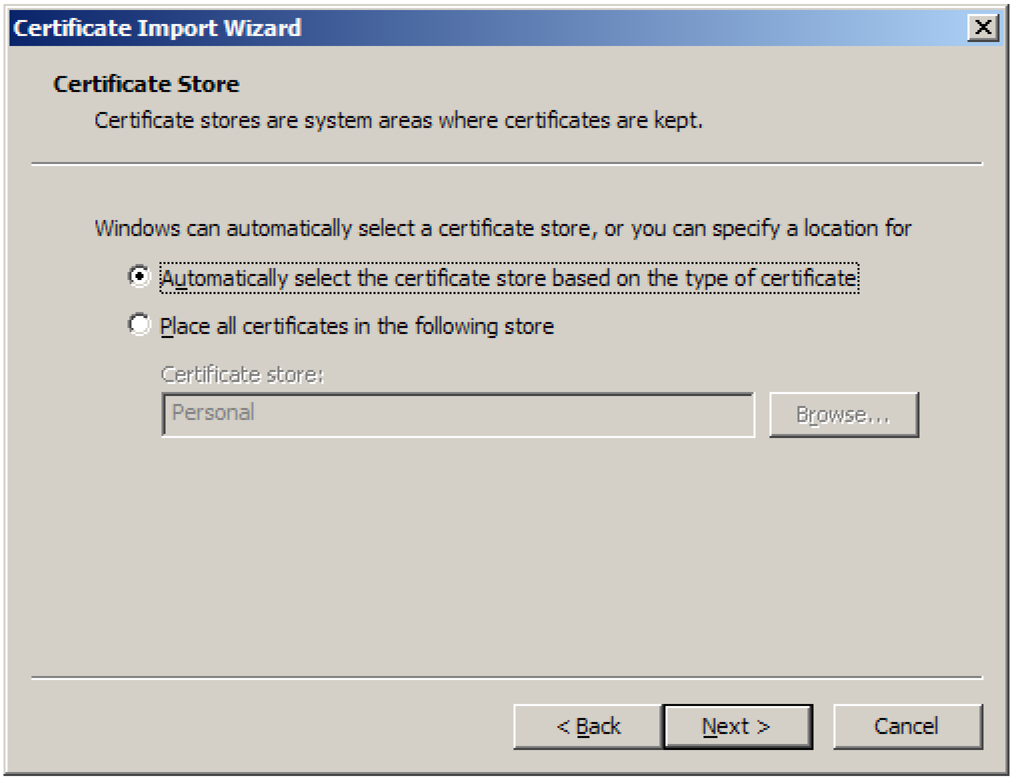
Personal Information Exchange- PKCS #12 (.PFX,.P12)

Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialized Certificate Store (.SST)

To continue, click the **Next** button.

9. **Internet Explorer – Certificate Import Wizard – Step 5**
   Performing the above action will invoke a window similar to the following:



Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for

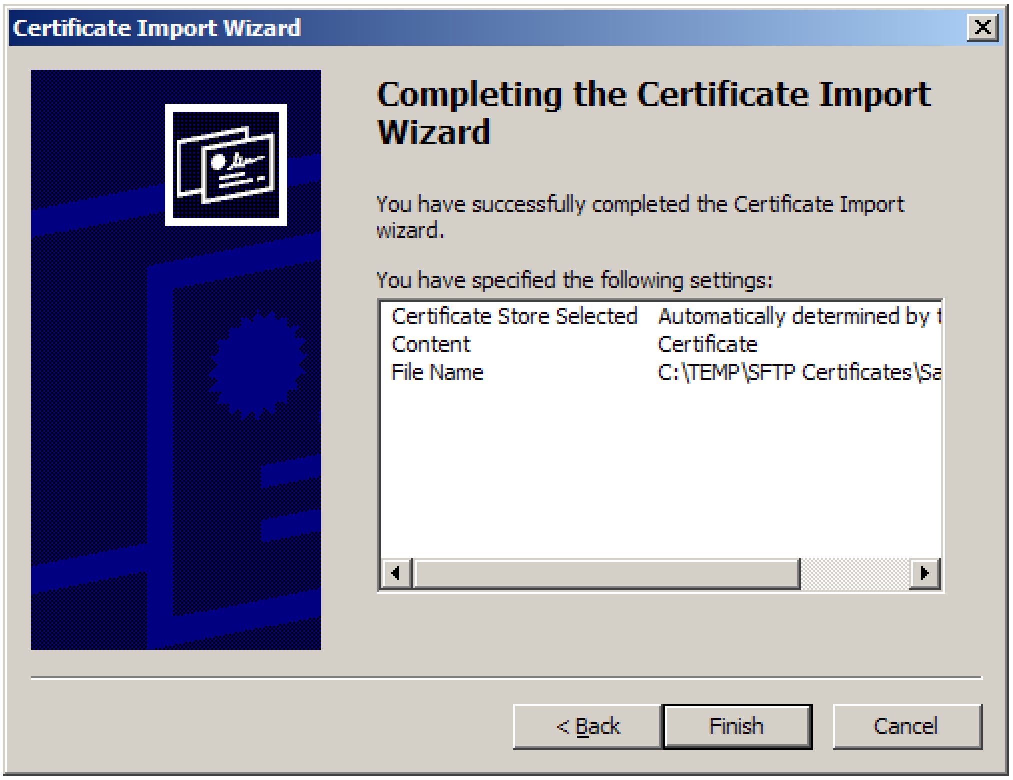
● Automatically select the certificate store based on the type of certificate

○ Place all certificates in the following store

Certificate store: Personal

To continue, first click the **Automatically select the certificate store based on the type of certificate** radio button (as shown above), then click the **Next** button.

10. **Internet Explorer – Certificate Import Wizard – Finish**
    Performing the above actions should now invoke a window similar to the following:



Completing the Certificate Import Wizard

You have successfully completed the Certificate Import wizard.

You have specified the following settings:

Certificate Store Selected	Automatically determined by t
Content	Certificate
File Name	C:\TEMP\SFTP Certificates\Sa

To continue, click the **Finish** button, after which all remaining open IE windows may be closed as desired.

If the above steps were followed successfully, the appropriate Certificate should now be loaded into the Windows Certificate Store (which is located within the Windows Registry), and Secured FTP communications using NatQuery against the remote FTP platform may occur.

Microsoft's WININET.DLL

To achieve FTP integration between NatQuery and the remote Natural / ADABAS server platform, NatQuery relies on the use of a Microsoft-provided Dynamic Link Library (DLL) called WININET.DLL.

WININET.DLL is a core component of the Microsoft Internet Explorer application, and as such it is typically installed onto a Windows workstation by default as a part of the Windows installation. If WININET.DLL is not available on the Windows workstation, this module can be easily obtained for free by installing the latest version of the Microsoft Internet Explorer application which is available from the Microsoft website.

To achieve successful integration between NatQuery and the remote Natural / ADABAS system, then not only must WININET.DLL be available on each client workstation, but the version of this DLL may be at least at a specific version level. The following requirements apply:

- For VSE and UNIX servers, WININET.DLL must be version 4.0 or greater.

- For MVS systems, WININET.DLL must be version 5.0 or greater to achieve "Direct FTP" integration. For "Just FTP" integration, then WININET.DLL must be version 4.0 or higher.

Without WININET.DLL available on a NatQuery workstation, no automated FTP integration can be achieved. To determine the version number of WININET.DLL on any Windows workstation, please refer to the section entitled Determining the version of WININET.DLL.

Determining the version of WININET.DLL

To determine what version of WININET.DLL is on your workstation, perform the following:

1. Click on the **Start** button on the Windows Task Bar.

2. Click on the **Find** menu item.

3. Click on the **Files and Folders** menu item.

4. On the "Find: All Files" window, enter "WININET.DLL" into the "**Named:**" textbox, and select "Local Hard Drives" in the "**Look In:**" drop-down list box. Now click the **Find Now** button.

5. Once the Find operation has finished, review the contents of the lower portion of the screen that should display the location of the WININET.DLL file. If the lower portion of the "Find: All Files" window displays Wininet.dll, then proceed to the next step.

   If Wininet.dll is not found – then you will not immediately be able to use NatQuery against your server environment and you should not proceed to the next step. You will need to obtain the Wininet.dll file from Microsoft, and one of the easiest ways to do this is to download the current version of Microsoft's Internet Explorer product.

6. If the file is found, then right-click on the WININET.DLL file and on the menu that presents itself click on the **Properties** menu item. This will present a "Wininet.dll Properties" window that should have two tabs at the top, **General** (which is presented by default) and **Version**.

7. Click on the **Version** tab, and take note of the version number displayed in comparison to the recommended versions shown above. If the current version is not correct, then you will need to obtain the Wininet.dll file from Microsoft, and one of the easiest ways to do this is to download the current version of Microsoft's Internet Explorer product.

Automatic Update of NatQuery Versions

In situations where NatQuery is utilized as an End-User data extraction tool, it is likely that there will exist at least one version of NatQuery installed as an Administrator version, and there will exist one or more installations of NatQuery installed as End-User versions.

From time to time, NatWorks will make available updated versions of NatQuery, with this software being provided on CD or freely downloadable from the NatWorks, Inc. website.

When new versions of NatQuery are released, it is strongly suggested that these new versions be applied to all installations of NatQuery that may exist within an organization.

Since NatQuery is a workstation application, the historical approach to upgrading workstation products is typically performed by having a new version of the given product physically installed onto each workstation. This approach is, at best, time consuming and unwieldy, and it may additionally be difficult to insure that all installations are in fact upgraded in a timely manner.

To alleviate this problem, NatQuery has the ability - in networked environments - to perform an “Automatic Update” of End-User version installations, and have this Automatic Update be initiated against all End-User machines from a single Administrative workstation.

To have NatQuery become enabled to handle Automatic Update, the following conditions must be met:

- An Automatic Update can only occur against a NatQuery installation that has had NatQuery 2.4.2 (or higher) installed.

- The NatQuery workstations must all exist on a common network, and be able to share a designated network path (directory).

- The NatQuery workstations must all resolve their Environment Configuration Path from a network path.

- The network path into which the Automatic Update files will be placed must either be READ accessible to all users, or each NatQuery user must be allowed READ permission to this network path.

- An Administrator version of NatQuery must exist on the network. Further, the network path into which the Automatic Update files will be placed must either be WRITE accessible to all Administrators, or each NatQuery Administrator must be allowed WRITE permission to this network path.

The basic process involved with an Auto Update scenario is as follows:

1. **New Version of NatQuery is Obtained**
An Administrator obtains a new version of NatQuery, and additionally obtains an

Automatic-Update file for this new version. This software will be periodically provided by NatWorks as updates, or can periodically downloaded from the NatWorks, Inc. website (http://www.natworks-inc.com).

2. **Administrator Installs New Version**
   The Administrator installs the latest version of NatQuery onto an administrative workstation.

3. **Automatic Update is Configured and Deployed**
   Using the **Automatic Update** function of NatQuery, the Administrator would Configure Automatic Update.

   With NatQuery started and with no query open on the desktop, this function is accessed by first clicking on the **Administer** drop-down menu and then clicking on **Automatic Update**. After the automatic update file has been specified by the Administrator, clicking on the **Deploy Update** button will deploy the update to the currently designated Environment Path.

   Specific Help on using the **Automatic Update** function is available by either hitting **F1** or by clicking on the **Help** button while in this function.

The above steps have the effect of building a "Trigger File" into the current Environment Path. Assuming that the setting of this path points to a networked path and this path is shared by other NatQuery workstations, then at the point in time when these other NatQuery installations are next executed, they will become aware of the existence of the latest available version by "seeing" this Trigger File.

Upon "seeing" the Trigger File, the individual workstation will then examine the contents of this file so as to compare the newly available version to the current version that is executing. If the available version is more recent, then NatQuery will notify the user of the existence of a new version – and will request approval to perform the update. If approval is given, then the .EXE file of the existing version of NatQuery will be replaced with the newer version.

NatQuery Dynamic Substitution Variable Reference Table

The following table summarizes the variables that NatQuery can make available for use in Production JCL / Script execution templates. Please note that in some cases specific dynamic variables may not be available for substitution within JCL / Script templates depending upon the JCL / Script involved and other settings within NatQuery.

Dynamic Tag	**Description**	**Where Set**
&&NETWORK-PASSWORD	Network User Password	NatQuery will prompt the user for the value of this field, which will contain any password required to access a Network resource. This substitution variable is almost always used with the substitution variable &&NETWORK-USER.
&&NETWORK-USER	Network User ID	NatQuery will prompt the user for the value of this field, which will contain a User ID that may be required to access a Network resource. This substitution variable is almost always used with the substitution variable &&NETWORK-PASSWORD.
&&USER-ID	This is the value of a NatQuery User's User ID, and is typically used to identify a NatQuery user against the Natural server platform.	**NatQuery Configuration, Server User-ID** field. This value is also used to match against users defined through **Administer User Information**
&&USER-NAME	User's Name	**Administer Users / Administer User Information** for specified user, **User Name** field.
&&USER-QUERY-PREFIX	User's Query Prefix	**NatQuery Configuration**, **General Defaults** tab, **Default Query Prefix**.
&&USER-QUERY-SLOT	User Query Slot	Internally provided by NatQuery, this is the relative request slot number the query was placed into.
&&USER-EMAIL-ADDRESS	User Email Address	**Administer Users / Administer User Information** for specified user. Only available if **Enable E-Mail Handling of Output Data** is selected on the **Miscellaneous** tab on the **Administer Server Connection Information** function.
&&USER-EMAIL-SUBJECT	User Email Subject	Provided to NatQuery by a user, this value equates to a request's **Query Name**. Only available if **Enable E-Mail Handling of Output Data** is selected on the **Miscellaneous** tab on the **Administer Server Connection Information** function.

Dynamic Tag	Description	Where Set
&&USER-EMAIL-MESSAGE	User Email Message	Internally provided by NatQuery. Only available if **Enable E-Mail Handling of Output Data** is selected on the **Miscellaneous** tab on the **Administer Server Connection Information** function.
&&USER-EMAIL-DIR-REF	User Email Directory	**Administer Server Connection Information, Directory References tab** – this value equates to the **Output Files Directory** (if one exists). Only available if **Enable E-Mail Handling of Output Data** is selected on the **Miscellaneous** tab on the **Administer Server Connection Information** function.
&&USER-EMAIL-FILE-REF	User Email File	**Administer User Information** – this value equates to user's **User Output File.** Only available if **Enable E-Mail Handling of Output Data** is selected on the **Miscellaneous** tab on the **Administer Server Connection Information** function.
&&USER-REQUEST-ID	User Request ID	Internally provided by NatQuery to be a unique ascending number which is unique to the user and which identifies a specific user request.
&&USER-NATURAL-LIBRARY	User Natural Library	Natural Server Information
&&USER-REQUEST-FILE	User Request File(s)	**Administer User Information**, **Request File(s)** tab for the current NatQuery user.
&&USER-OUTPUT-FILE	User Output File(s)	**Administer User Information**, **Output File(s)** tab for the current NatQuery user.
&&USER-LOG-FILE	User Log File	**Administer User Information**, **Log File** tab for the current NatQuery user.
&&USER-ACCT	User Account	**Administer User Information**, **Accounting** tab for the current NatQuery user.
&&USER-SUBACCT	User Sub-Account	**Administer User Information**, **Accounting** tab for the current NatQuery user.
&&USER-PROGRAM-NAME	User Program Name	Provided to NatQuery by a user, this value equates to a request's **Program Name**.
&&USER-SPECIAL-PROCESS	User Special Process	Internally provided by NatQuery to support the commands required to execute a SYSTRANS request. Generally this is used exclusively in the **Production Special Process** JCL template.

Dynamic Tag	Description	Where Set
&&USER-EXTRACT-PROGRAM	User Extract Program	Internally provided by NatQuery, this tag will be replaced by a generated Natural program. Should be used exclusively in the **Production Request Process** JCL template.
&&SERVER-PARAMETER	Server Parameter	This parameter should be used exclusively in the **Production Request Server** template. This value is hard coded in the initial job that starts the server, and is dynamically substituted by the server as it processes. This parameter relates to the **NatQuery Server**, which is now obsolete.
&&OUTPUT-LRECL	Output File LRECL	Internally built by NatQuery, this value contains the logical record length that is calculated by NatQuery for a specific **User Output File**.
&&OUTPUT-EST-RECORDS	Output Estimated Records	Internally built by NatQuery, this value contains the estimated number of records returned by a NatQuery extract process, and is associated with a specific **User Output File**.
&&OUTPUT-UNITS	Output Units	Internally built by NatQuery using **Disk Size Constants** data from the **Administer Server Connection Information** function.
&&OUTPUT-BLKSIZE	Output Blocksize	Internally calculated by NatQuery using information based on the calculated **LRECL** and **Disk Size Constants** data from the **Administer Server Connection Information** function.
&&OUTPUT-PRIMARY	Output Units Primary	Internally calculated by NatQuery based on **Output Estimated Records, Output Block size**, and the **Primary Disk Allocation Factor** found on the **Disk Size Constants** tab of the **Administer Server Connection Information** function. This is the primary number of **Output Units** to be allocated for a given **User Output File**.
&&OUTPUT-SECONDARY	Output Units Secondary	Internally calculated by NatQuery based on **Output Estimated Records, Output Block size**, and the **Secondary Disk Allocation Factor** found on the **Disk Size Constants** tab of the **Administer Server Connection Information** function. This is the secondary number of **Output Units** to be allocated for a given **User Output File**.

Dynamic Tag	Description	Where Set
&&JOB-CLASS	Job Class	**Administer Server Connection Information**, **Job Priority** tab; used in conjunction with the **High Priority Execution / Low Priority Execution** options on the **Send to Server Option** window. This value is optional and should only be used in the **Production Request Process** and **Production Special Process** JCL templates
&&JOB-CANCEL-MESSAGE	Job Cancel Message	Internally provided by NatQuery, this value is passed to the execution of **NQYP0002 / NQYPNT06.**
&&NATCDC-PARMS	NatCDC Parameter File	**Generate NatCDC Objects**, **File Naming** tab, **NatCDC Parameter** field. This file will contain the file-specific parameters required by the NatCDC module. Only available if NatQuery is provided with a NatCDC License Key.
&&NATCDC-ADASEL	NatCDC ADASEL Output	**Generate NatCDC Objects**, **File Naming** tab, **ADASEL Output** field. This file will contain the file-specific output from an execution of ADASEL. Only available if NatQuery is provided with a NatCDC License Key.
&&NATCDC-INTRM1	NatCDC Intermediate File 1	**Generate NatCDC Objects**, **File Naming** tab, **NatCDC Intermediate-1** field. This file will contain the file-specific, fixed length output of the NatCDC module. Only available if NatQuery is provided with a NatCDC License Key.
&&NATCDC-INTRM2	NatCDC Intermediate File 2	**Generate NatCDC Objects**, **File Naming** tab, **NatCDC Intermediate-2** field. This file will contain the file-specific, fixed length output of the system SORT program. Only available if NatQuery is provided with a NatCDC License Key.
&&NATCDC-INTRM-LRECL	NatCDC Intermediate Logical Record Length	Internally calculated by NatQuery, this value is the logical record length of &&NATCDC-INTRM1 and &&NATCDC-INTRM2. Only available if NatQuery is provided with a NatCDC License Key.
&&NATCDC-INTRM-BLKSIZE	NatCDC Intermediate Blocksize	Internally calculated by NatQuery, this value is the block size of &&NATCDC-INTRM1 and &&NATCDC-INTRM2. Only available if NatQuery is provided with a NatCDC License Key.

Dynamic Tag	Description	Where Set
&&NATCDC-FINAL-LRECL	NatCDC Final Record Length	Internally calculated by NatQuery, this value is the Blocksize of &&NATCDC-FINAL. Only available if NatQuery is provided with a NatCDC License Key.
&&NATCDC-FINAL-BLKSIZE	NatCDC Final Blocksize	Internally calculated by NatQuery, this value is the Blocksize of &&NATCDC-FINAL. Only available if NatQuery is provided with a NatCDC License Key.
&&NATCDC-FINAL	NatCDC Final File	**Generate NatCDC Objects**, **File Naming** tab, **NatCDC Final Output File** field. This file will contain the file-specific, fixed length final output of the NatCDC process. Only available if NatQuery is provided with a NatCDC License Key.
&&NATSEC-USER-ID	Natural Security User ID	If the **Natural Security Installed** option is enabled in **Natural Server Information** function, then NatQuery will prompt for a Natural Security User ID when a request is sent to the server – the value captured will be substituted for this value.
&&NATSEC-USER-PWD	Natural Security User Password	If the **Natural Security Installed** option is enabled in **Natural Server Information** function, then NatQuery will prompt for a Natural Security password when a request is sent to the server – the value captured will be substituted for this value.
&&SQL-FILENAME	Filename prefix for use with RDBMS integration to establish consistent naming	Internally set by NatQuery to the value of: &&SQL-USER-ID "_" &&PROGRAM-NAME "_REQ" &&USER-REQUEST-NUMBER
&&SQL-DATA-FILE	File name of data file as placed on the RDBMS machine	Internally set by NatQuery
&&SQL-SERVER-PATH	Server's Path to the Server directory where RDBMS files will be resolved from	Specified on Integration Setup
&&SQL-ISQL-PATH	Path to the ISQL utilities directory (local or remote)	Specified on Integration Setup
&&SQL-BCP-PATH	Path to the BCP utility directory (local or remote)	Specified on Integration Setup
&&DATABASE-NAME	Name of the target database on the RDBMS server	Specified on Integration Setup

Dynamic Tag	Description	Where Set
&&STAGING-TABLE-NAME	Name of the staging table that will be created on the RDBMS server platform	Internally set to: &&PROGRAM-NAME "_STAGING"
&&SQL-SERVER-NAME	Machine name of the RDBMS server	Specified on Integration Setup
&&SQL-CLIENT-PATH	Client's Path to the Server directory where RDBMS files will be resolved from	Specified on Integration Setup
&&SQL-USER-ID	User ID of the User as identified to the RDBMS machine	NatQuery will prompt the user for this value if referenced
&&SQL-PASSWORD	Password that is associated with the User ID of the User as identified to the RDBMS machine	NatQuery will prompt the user for this value if referenced

NatQuery Internal File Reference

For reference purposes, the following table outlines the internal files that NatQuery either utilizes or creates, and details how these files are created. Unless otherwise noted in the Description column, all files listed are created through or manipulated by Administrator functions and are a part of an Environment Configuration.

File Name	Description	Where Created / Modified
*.NSD	DDM File	Import DDM / FDT
VALIDENV.CFG	Valid Environment Configuration file, encrypted	Created as a result of executing the Verify Environment Configuration function
FILEDESC.CFG	Contains Descriptor Statistic Information for DDM files in an Environment Configuration	Administer Descriptor Statistics
FILEOCCS.CFG	Contains Occurrence Information for DDM files in an Environment Configuration	Administer Occurrence Information
FILERELS.CFG	Contains File Relationship Information for DDM files in an Environment Configuration	Administer File Relationships
xxxxJCLU.PRD	Production Special Process JCL / Script template (where “xxxx” is either “MVS”, “VSE” or “UNIX”)	Administer FTP / JCL Script, Production Special Process JCL / Script template
xxxxJCLC.PRD	Production Request Cleanup JCL / Script template (where “xxxx” is either “MVS”, “VSE” or “UNIX”). Not required for Direct FTP.	Administer FTP / JCL Script, Production Request Cleanup JCL / Script template
xxxxJCLS.PRD	Production Server Process JCL / Script template (where “xxxx” is either “MVS”, “VSE” or “UNIX”). Not required for Direct FTP.	Administer FTP / JCL Script, Production Server Process JCL / Script template
xxxxJCLR.PRD	Production Request Process JCL / Script template (where “xxxx” is either “MVS”, “VSE” or “UNIX”)	Administer FTP / JCL Script, Production Request Process JCL / Script template
SERVER.CFG	Contains information pertaining to the remote Natural environment	Natural Server Information
FTPSERVR.CFG	Contains Information pertaining to how FTP is processed against the remote server	Administer FTP Server Information

FTPUSERS.CFG	Contains information pertaining to all defined FTP users of NatQuery	Administer FTP User Information
FILESIGN.CFG	Contains information pertaining to File Sign handling for all fields in all files (DDMs) that require a sign byte.	Administer Sign Byte Information
AUTOUDP.CFG	Contains Automatic Update Information	Automatic Update
TRIGGER.NEW	Contains Automatic Update Version Information	Automatic Update
*.AUD	Contains Automatic Version to be rolled out	Provided by NatWorks either directly or from their website
CONTROL.TXT	Generated file that contains the information that should be populated into the Server Control File. Only used when NatQuery is configured to use Just FTP.	FTP Send Control File to Server
NQYP000x.TXT	NatQuery Processing Program. NQYP0001.TXT represents the Natural program that will execute the NatQuery Server program when NatQuery is used with Just FTP. NQYP0002.TXT and NQYP0003.TXT are both NatQuery messaging programs	Provided by NatQuery installation
CONFIG.CFG	Contains Information related to a specific installation of NatQuery. This file will always reside in the FILES sub-directory of the NatQuery install directory for each NatQuery installation.	NatQuery Configuration
*-FTP.LOG	FTP interaction log files. These files are built in response to NatQuery FTP operations to assist in the potential debugging of problems with FTP. These files will have a file name that equates to the submitting user's User-ID with "-FTP" appended, and will have a "LOG" extension. This log covers only the last FTP interaction of a given user.	Built By NatQuery in response to user FTP operations
*.STS	DDM Descriptor Statistics	Built by NatQuery once a Descriptor Statistics program has been executed and then downloaded.

*.QRY	Query extract specifications	Built, modified or deleted as a result of a user building an extract specification (query).
*.EXM	Example JCL / Script templates	Provided by NatQuery installation
*.JCL	User execution request files that have been sent to the server for processing. This means that these files will contain both a generated program / request, with the program imbedded into the JCL / script to execute it. These files will have a file name that equates to the submitting user's User-ID with a "JCL" extension, and will be the last execution request submitted.	Generated by NatQuery as a result of a Send To Server request
*.LLG	User-specific Local Log Files. This file will always reside in the FILES sub-directory of the NatQuery install directory on the NatQuery workstation. These files will have a file name that equates to the submitting user's User-ID with a "LLG" extension, and will exist if the user has submitted any request.	Generated by NatQuery as a result of a Send To Server request
*.RLG	User-Specific Remote Log Files. These files will be placed into the Environment Configuration path when a user requests a Check Server for Update – this file is the contents of the User's Remote Log File. These files will have a file name that equates to the submitting user's User-ID with a "RLG" extension.	Built by an FTP Get operation performed by NatQuery in response to a user's request a Check Server For Update.
LICENSE.KEY	This file contains the License Key for an individual installation of NatQuery. This file will always exist in the FILES subdirectory of the NatQuery install directory on a NatQuery workstation.	Built By NatQuery and updated by providing License Key information to a NatQuery installation through the NatQuery Configuration function
*.NSP	NatQuery Generated Natural	Built by NatQuery generation

	Programs	functions

Troubleshooting

This section describes suggested actions to be taken in the event of problems being encountered with the use of NatQuery.

Depending upon the nature of the problem encountered, please refer to one of the following sections.

- Common NatQuery Error Messages**,**
- FTP Issues**,** and
- Job Control Language (JCL) / Execution Issues**.**

Common NatQuery Error Messages

This section details some of the more common NatQuery Error Messages, and provides information on the cause of these errors and how these error messages may be avoided / corrected. In most cases, the error messages produced by NatQuery are hoped to be at least somewhat self-explanatory as to the cause and resolution. The following list then is by no means a complete listing of all errors that NatQuery can produce. If a NatQuery-produced error message is seen, then it is advisable to first review this section for an answer before reviewing any other troubleshooting section.

The following Common NatQuery Error Messages are detailed in this section:

- Un-Verified of Non-Existent Configuration Information
- User ID not found in administratively provided User File
- User ID not designated as “Active”
- The Administratively-provided User File was not found
- NatQuery does not have a valid license key
- The trial version of NatQuery has expired
- An error occurred while verifying the current Environment
- An error occurred while initializing the current Environment
- An error occurred while verifying the DDM files
- An error occurred while verifying the FTP files
- No “Open” Request Slots

Un-Verified of Non-Existent Configuration Information

This error message is produced in the situation where the current setting of Environment Path on the NatQuery workstation either points to:

an invalid directory path
a directory path that contains no Environment Configuration information
a path that contains only partial Environment Configuration information
a path that does contain Environment Configuration information but this configuration has not yet been verified
a path in which one or more of the files' date and timestamps do not match the information stored in VALIDENV.CFG.

To correct this problem, the user should first verify that the current setting of the **Environment Path** on their NatQuery workstation does in fact point at the correct Environment Configuration path. This path can be seen by examining the contents of the Environment Path field located on the Environment Paths tab of the NatQuery Configuration function. If the path shown is incorrect, then this path should be modified so that it does point at the correct path.

If the **Environment Path** information is correct and NatQuery is operating in End-User mode, then the only alternative to correcting this problem is to either change the Environment Configuration path to point at a valid path, or to import (**Administer / Environment Configuration / Import Environment Configuration**) a valid Environment Configuration into the current path.

If the **Environment Path** is correct and NatQuery is operating in Administrator mode, then the cause of the problem is likely to be an incomplete Environment Configuration. By executing the **Verify Environment Configuration** function and reviewing the corresponding report, the Administrator can determine what is missing from the Environment Configuration, supply this required information, and then run the Verify Configuration function again.

User ID not found in administratively provided User File

This error message is caused when a user attempts to use NatQuery's automated FTP functionality to send a request to the server, and the current value of the user's User-ID is not matched with a User-ID provided by the Administrator through the Administer FTP User function.

To correct this problem, the user should first verify the current value of the User-ID specified to their NatQuery workstation is correct. The current setting of the User-ID can be seen on the User Identification tab of the NatQuery Configuration window. This will invoke the NatQuery Configuration window, and the current value of the User-ID that NatQuery is operating under will be found on the User Identification tab of this window.

If the displayed value is correct, then the Administrator must then verify that this User-ID is present in the list of FTP users as built through the FTP User Information function. This function is invoked on a NatQuery workstation operating in Administrator mode by clicking on Administer, then clicking on Environment Configuration, then clicking on FTP Configuration, and then clicking on FTP User Information. Using the functionality of the Administer FTP User window, the Administrator should verify that there is an entry for the specific User-ID (if not then one should be added), that the spelling and case of the two User-ID's match (if they do not then one or the other should be changed so that they do match), and that the User-ID is designated as "active" in the Administer FTP User function.

User ID not designated as "Active"

This error message is caused in the situation where the current NatQuery user's User ID was properly matched against an entry in the FTP User configuration file, but the Administer FTP User information indicates that the User ID is not 'Active'.

To correct this problem, the Administrator must invoke the Administer FTP User function (Administer / Environment Configuration / FTP Configuration / FTP User Information), select the user of interest, modify the status of this user to be 'Active', click Modify to store the user information, then run the Verify Environment Configuration.

The Administratively-provided User File was not found

This error is caused when the file FTPUSERS.CFG cannot be located in the directory specified by the current Environment Configuration path of the NatQuery workstation.

The most likely cause of this situation is an incorrect setting of the Environment Configuration path for the workstation producing this error. An incomplete Environment Configuration, or an Environment Configuration that does not contain complete FTP configuration information can also cause this.

The suggested solution is to first verify that the current setting of the workstation's Environment Path is correct. This path can be seen on the Environment Paths tab of the NatQuery Configuration window.

If the path is incorrect, then this path should be changed so that it points to a path that contains a verified Environment Configuration.

If the path is correct, then the Administrator should verify that FTP information exists in this path, should build this information if it is not, and should then verify the environment configuration paying particular attention to the FTP portion of the Verify Configuration report.

NatQuery does not have a valid license key

This error message will be displayed at NatQuery startup if NatQuery determines that there is no License Key information available to NatQuery.

To be able to operate in any mode other than "Demo" mode, NatQuery must be given a License

Key. To obtain a valid License Key, you must contact NatWorks, Inc. Information on contacting NatWorks can be found in the section entitled Contacting NatWorks.

The trial version of NatQuery has expired

This message will be displayed at startup when NatQuery determines that the License Key it has been given is a “trial” license key, and the period of time associated with this “trial” has now expired.

To be able to operate in any mode other than “Demo” mode, NatQuery must be given a License Key. To obtain a valid License Key, you must contact NatWorks, Inc. Information on contacting NatWorks can be found in the section entitled Contacting NatWorks.

An error occurred while verifying the current Environment

This message may occur when NatQuery determines there are differences between the configuration data and timestamps stored in the file VALIDENV.CFG (created as a result of executing a Verify Environment Configuration) and the current date and timestamps in one or more of the configuration files.

In a version of NatQuery that is executing in End-User mode, there is no solution to this other than either importing a new Environment Configuration (assuming that the installation is not networked to use a network Environment Configuration path), or to notify the Administrator that a Verify Configuration must be run (in the case where a network Environment Configuration is being used).

In a version of NatQuery that is operating in Administrator mode, the solution to this problem is to rerun a Verify Configuration.

An error occurred while initializing the current Environment

This message occurs when NatQuery attempts to open internal configuration files that should exist in the current Environment Configuration path, but there are problems encountered when these files are opened or these files no longer exist.

In a version of NatQuery that is executing in End-User mode, there is no solution to this other than either importing a new Environment Configuration (assuming that the installation is not networked to use a network Environment Configuration path), or to notify the Administrator that a Verify Configuration must be run (in the case where a network Environment Configuration is being used).

In a version of NatQuery that is operating in Administrator mode, the solution to this problem is to rerun a Verify Configuration.

An error occurred while verifying the DDM files

This error occurs when NatQuery attempts to match the date and timestamp of a "verified" DDM to the current date and timestamp of the DDM and a difference is found.

In almost all cases, this indicates that either the DDM was manually changed, or that the DDM has been recently imported without a Verify Configuration being executed.

In a version of NatQuery that is executing in End-User mode, there is no solution to this other than either importing a new Environment Configuration (assuming that the installation is not networked to use a network Environment Configuration path), or to notify the Administrator that a Verify Configuration must be run (in the case where a network Environment Configuration is being used).

In a version of NatQuery that is operating in Administrator mode, the solution to this problem is to rerun a Verify Configuration.

An error occurred while verifying the FTP files

This error occurs when NatQuery attempts to match the date and timestamp of a "verified" FTP configuration file to the current date and timestamp of the FTP configuration file and a difference is found.

In almost all cases, this indicates that the FTP configuration was manually changed, the FTP configuration file was changed through NatQuery, or that the FTP configuration file was recently imported without a Verify Configuration being executed.

In a version of NatQuery that is executing in End-User mode, there is no solution to this other than either importing a new Environment Configuration (assuming that the installation is not networked to use a network Environment Configuration path), or to notify the Administrator that a Verify Configuration must be run (in the case where a network Environment Configuration is being used).

In a version of NatQuery that is operating in Administrator mode, the solution to this problem is to rerun a Verify Configuration.

No "Open" Request Slots

NatQuery will produce a message similar to the above in any situation where the user is attempting to submit a request, but all of the user's existing request "slots" are filled with other requests.

Each request submitted by a NatQuery user will fill an open request "slot" that has been assigned to the user by the NatQuery Administrator. When all request "slots" are filled with pending requests, NatQuery will not be able to place any new requests into an "OPEN" slot until one of the existing request slots are released, or an existing request slot with a "PENDING" status is manually cleared.

Usually, request slots are cleared automatically by NatQuery when a user invokes the Check Server function, clicks the Check Server for Update button, and then instructs NatQuery to Retrieve or Clear the output of any request that is marked “DONE”. This process should result in a request slot that was previously marked “PENDING” / “DONE” being changed to “OPEN”. This is an important aspect of NatQuery’s Request Slot handling: Request Slots are “re-used” over time.

In other cases, particularly when NatQuery responds to a Check Server for Update request and returns the fact that one or more requests have “FAILED”, then the user must manually clear “FAILED” requests by first clicking anywhere within the text associated with the “FAILED” request and then clicking the Clear Selected Request button. This will cause NatQuery to update the specific request to have a status of “OPEN”.

In still other cases, such as what is likely to occur when initially configuring NatQuery, it is probable that situations will arise that cause a request to be submitted however the execution of the requests fails, and this failure notice is not properly logged into the specific user’s Remote Log file. With this situation, the “PENDING” requests have essentially been orphaned – and the user must manually clear the “PENDING” request that is known to have failed. This is accomplished by first clicking anywhere within the text associated with the “PENDING” (FAILED) request and then clicking the Clear Selected Request button. This will cause NatQuery to update the specific request to have a status of “OPEN”. Prior to just deleting any request that is “PENDING”, the user should attempt to insure that the request is not simply queued up for execution or had otherwise simply not yet been executed.

FTP Issues

This section details how to proceed should NatQuery encounter problems with FTP broken out in the following categories:

- NatQuery FTP Interaction Reference,
- FTP Log File,
- Testing FTP, and
- FTP Error Messages.

NatQuery FTP Interaction Reference

When Interfacing with FTP, NatQuery utilizes information that has been created by an Administrator as well as information entered by a user to perform FTP related tasks. This section details the FTP-related information that is stored within NatQuery, as well as the functions which allow this information to be manipulated.

- **Communication Mode**
  The **Communication Mode** of NatQuery is set using the Natural Server Information function, with the choice of communication mode being "**FTP**", "**PC Network**" or "**none**".

  To use **FTP** or **PC Network** as a Communication Mode, this option is set from an empty NatQuery desktop by clicking on **Administer / Environment Configuration / Natural Server Information; t**he Natural Server Information function presents a selection box called **Server Communication Mode**.

  It should be noted that **FTP** can be used against virtually any Natural / ADABAS source environment (as long as that environment is running an FTP Server), whereas **PC Network** can only be used against environments that can share disk with the NatQuery environment (Windows, UNIX or Linux).

- **FTP IP Address / URL**
  The IP Address / URL of the FTP Server (which will also generally be the platform that Natural resides on), is set by the Administrator through the **Server Connection Information** function. From an empty NatQuery desktop, this function is accessed by first clicking on **Administer**, then clicking on **Environment Configuration / Server Connection Configuration / Server Information**.

  By default the **Server Information** function will present the **Connection Info** tab which controls the **Serve Name** field that allows for the capture of the IP Address or URL.

- **FTP User ID**
  The **User Id** value used for FTP connections will be the value of the **User ID** as entered by the User on the **NatQuery Configuration** function. The **NatQuery Configuration** function is accessed from an empty NatQuery desktop by clicking on **Administer** and then clicking on **NatQuery Configuration**. By default the **User Identification** tab will be displayed when the **NatQuery Configuration** function is presented; this will be the User ID used by NatQuery when it makes an FTP connection.

  This User ID value must *exactly* match (User Ids are case-sensitive) a User ID created by the Administrator using the NatQuery **Administer User** function. The **Administer User** function is accessed from an empty NatQuery desktop by clicking on **Administer / Environment Configuration / Server Connection Configuration / User Information**.

Of course, the value of the FTP User ID must also be defined to the FTP Service itself.

- **FTP User Password**
  The Password value will be prompted for by NatQuery the first time it requires FTP within a single NatQuery session. Subsequent to this, the Password is stored in memory and re-used as needed; this Password value will be discarded when NatQuery is terminated (it is not stored by NatQuery).

  If a Password value is entered incorrectly, then this fact will likely be reported back to the user via an FTP Error message – and upon the next invocation of an FTP service by NatQuery; the Password value will be prompted for again.

  Of course, the Password value must precisely correspond to a User ID / Password combination defined to the FTP Service itself.

- **FTP Working Directory**
  When a FTP session is invoked against a remote server, the FTP service running on that server may or may not establish a "working directory" by default. In some cases, such as when FTP is used against MVS or VSE and Direct FTP is used – then the concept of a working directory will not apply to files being placed on the server since NatQuery will typically be FTPing into JES or POWER respectively. It will still apply however to where files are placed by an executed JCL stream, such that FTP can retrieve them. It should be noted that the concept of FTP working directories equates to high-level qualifiers when an FTP service is running on most mainframe platforms.

  If the default working directory established by the FTP service will align with how the Administrator would like NatQuery-related files to be placed on the server, then the option **FTP Session Establishes Working Directory** should be set in NatQuery. Alternatively, if the default working directory established by the FTP service does not align with how the Administrator would like to NatQuery-related files to be placed on the server or this placement would otherwise be against site standards, then the **FTP Session Establishes Working Directory** option should NOT be set, and the **Directory References** options of NatQuery should be used instead. With the **Directory References** options set with a value(s), NatQuery will perform an FTP Change Directory ("**CD**") operation to the named directory / qualifier for a specific object prior to performing the associated PUT or GET.

  In this way, the **FTP Session Establishes Working Directory** option takes precedence over the **Directory Reference** option(s); if **FTP Session Establishes Working Directory** is set / checked, then NatQuery will completely ignore **Directory References**, even if these have values supplied. If **FTP Session Establishes Working Directory** is not set / left un-checked, and the **Directory Reference** associated with the affected object has a value, then this value will be used in conjunction with a NatQuery-issued Change Directory ("**CD**") command.

  Both the **FTP Session Establishes Working Directory** option and the **Directory**

References options are set through the **Server Connection Information** function. This function is accessed from an empty NatQuery desktop by first clicking on **Administer**, then clicking on **Environment Configuration / Server Connection Configuration / Server Information**. Once in this function, the Session **FTP Establishes Working Directory** option is set through the **Miscellaneous** tab, and the **Directory References** options are set through the **Directory References** tab.

Directory References can be individually applied to **Request Files**, **Output Files** or **Log Files**.

- **FTP File Names**
  NatQuery will typically always manage the naming of **Request Files** and **Log Files**, and in most cases will also assign the names for **Output Files** (although users can assign their own names to output files under certain circumstances).

  The naming for **Request Files**, **Output Files** and **Log Files** are assigned to each user by an Administrator using the **Administer User** function of NatQuery. This function is accessed from an empty NatQuery desktop by clicking on **Administer / Environment Configuration / Server Connection Configuration / User Information**.

  While file *names* are specified through **Administer User** function, it is important to note that the directory or high-level qualifier under which these files will be created may be influenced by **FTP Working directory** settings as outlined immediately above.

- **Direct FTP / Just FTP**
  For NatQuery, the term **Direct FTP** applies to the situation where NatQuery is expected to facilitate the execution of am FTPed request made by a user.

  When used against mainframes, **Direct FTP** typically means that an FTP operation will pick up a NatQuery-generated request from the NatQuery workstation and then place this request (a generated JCL stream) directly into JES (for MVS) or POWER (for VSE). By placing a JCL stream directly into JES / POWER; automatic execution of a NatQuery generated request can be easily achieved. This contrasts against the use of **Just FTP** against mainframes, where an FTP simply moves a request into a file on the server platform (where manual execution of the request can then occur, or possibly automated execution outside the scope of NatQuery).

  For **Direct FTP** against MVS mainframes, this means that NatQuery must (and will) issue a special FTP command of "SITE FILETYPE=JES" (the default is usually FILETYPE=SEQ), this switches the FTP service to point at JES handling as opposed to typical sequential file movement. For **Direct FTP** against VSE systems, this means that NatQuery will adjust the target of the FTP to be "power.rdr.***x***", where "***x***" is the job class being submitted into.

  When used against Windows, UNIX or Linux servers **Direct FTP** typically means that NatQuery will use FTP to place a generated request into file(s) on the server, and then

NatQuery will perform a **Remote Execution Command** (such as RSH, SSH, or REXEC) to execute that specific request.

The choice of **Direct FTP** versus **Just FTP** is made on the **Server Connection Information** function. From an empty NatQuery desktop, this function is accessed by clicking on **Administer**, and then clicking on **Environment Configuration / Server Connection Configuration / Server Information**. The Request Submission options frame shown on the Connection Info tab (displayed by default) controls the setting of **Direct** versus **In-Direct FTP**.

In those situations where the server platform is Windows, UNIX or Linux is the server platform and **Direct FTP** is used, then the execution of a submitted request will be dependent upon a **Remote Execution Command** issued by NatQuery. The command that will be issued by NatQuery to accomplish this is configured through the use of the **Remote Execution Command String**. The command string value to use, which is subject to the use of dynamic substitution variables, is controlled through the **Server Connection Information** function's **PC Network / UNIX** tab. From an empty NatQuery desktop, this function is accessed by clicking on **Administer**, then clicking on **Environment Configuration / Server Connection Information / Server Information**. By default, the **Server Connection Information** function will present the **Connection Info** tab; the **Remote Execution Command String** is accessed on the **PC Network / UNIX** tab.

FTP Log File

To assist the Administrator in the troubleshooting of FTP related issues, the reader should be made aware that NatQuery creates a “log” file of FTP interaction(s) that may prove useful in determining what the cause of any given FTP problem – this log is written when an unexpected FTP error occurs.

NatQuery creates this FTP log file for each user when any FTP process is invoked, with this file being named as “***userid***-FTP.LOG” (where “***userid***” is the value of the current user’s User-ID as specified through the NatQuery Configuration function), and being placed into the current Environment Configuration path (as specified through the **NatQuery Configuration** function, **Environment Paths** tab). By reviewing the contents of this file after an FTP operation did not operate as expected, the Administrator can gain insight into the cause of the FTP problem.

Testing FTP

Perhaps an even greater asset to pinpointing problems with FTP involves the manual use of FTP. If NatQuery encounters problems with FTP, then in most cases these problems deal with the FTP service itself – not with NatQuery's use of the FTP service. Testing FTP connectivity manually then can often remove NatQuery as a probable cause.

Please note however that in order to manually test the FTP connection, the user will need the IP address or URL of the Natural server platform, a User-ID value that should work with FTP, along with the corresponding Password for this User ID.

To terminate an FTP interaction at any time, hold down the "**Ctrl**" key and then tap the "**C**" key. To close an FTP session from an FTP command prompt ("**ftp>**"), use the "**bye**" command.

To manually test FTP, the following general steps outline a suggested approach.

1. **Start FTP**
   This is most easily done by clicking the Windows **Start** button, then clicking **Run**.

   On the **Run** window, type:

   ftp

   into the **Open** textbox and then hit **Enter** or click the **OK** button.

   A Command prompt window will be opened, with a command prompt displayed that will be similar to:

   ftp>

2. **Issue OPEN command**
   At the FTP command prompt ("ftp>"), type:

   open *ip-address_or_URL*

   and then hit enter. For the value of *ip-address_or_URL*, the user will enter either the IP address of the Natural Server platform, or alternatively the URL name of the Natural server platform.

   Examples for an open command might be:

   open 192.168.0.2

   or

   open treebeard

If FTP responds with a message similar to "Unknown Host", then the value of the IP address or URL is most likely invalid, there is no TCP/IP connectivity to the desired machine, the FTP service is not running on the needed machine, or similar issues. If NatQuery is encountering FTP issues, and it has been given the same information as was used to manually test FTP – then the fault is not within NatQuery, and IT staff should be consulted.

How a FTP server will respond to an open command is dependent upon how the FTP service is configured. In most cases, there will be at least a two line response, with the first response line beginning with the number "**220**" followed by some text that usually identifies the FTP server.

The last line of the response will generally begin with the text "**User**" or possibly "Username". Either of these prompts may be by themselves, included with other text, and possibly end with a colon (":"). If this is the case, then the following step can be referred to. If this is not the case, then there will in all likelihood be an FTP error, and IT staff should be consulted.

3. **Provide User ID and Password**
   In response to FTP prompting in some way for a user, type:

   User-ID

   and hit enter, where ***User-ID*** is the value of a User-ID that should be able to use FTP services.

   In most cases, and regardless of what was enter for the user value, FTP should return a response line similar to:

   Password:

   In response to this password prompt, the valid password should be entered, with the Enter key then being hit.

   If the User and Password combination was entered correctly, and this User and Password combination are validly defined to the FTP Service, then FTP will typically respond with a line that begins with the number "**230**", followed by some text that should indicate a successful login. If this is the case, then the IP address, and the User ID / Password combination are not the problem with NatQuery's use of FTP.

4. **Test FTP File Movement / Handling**
   The act of being able to establish an FTP connection as described in the preceding steps is generally enough to pinpoint any FTP issue that NatQuery may be having. Meaning: If NatQuery was encountering problems with establishing an FTP connection to a given IP address or URL, or NatQuery was encountering issues with using a specific User ID

and Password combination – then the above mentioned steps should isolate where the problems lies (an invalid IP or URL, or an invalid User-ID Password combination).

For help on using FTP; the command "**help**" entered at the ftp prompt ("ftp>") will provide the commands that are available to FTP. To obtain help on a specific command, enter "**help *command-name***" at the ftp prompt.

While it is beyond the scope of this document to teach a user how to use manual FTP, the following information is provided to assist the reader is possible further testing of FTP.

Command	Description	Example
CD	Change Directory on server	cd *wkdir*
PUT	Place file on server	put *localfilename remotefilename*
GET	Retrieve file from server	get *remotefilename localfilename*

FTP Error Messages

In all cases with FTP errors, NatQuery attempts to provide terms of providing text that may be useful in determining be noted however that NatQuery internally makes calls to a library module provided by Microsoft with Internet

Further information on NatQuery use of WININET.DLL can be found in the section entitled Configure NatQuery to use FTPS against a RDBMS Target

This section describes the steps needed to configure NatQuery to use a Secure FTP connection against the platform upon which a target RDBMS resides. Depending upon how your organization uses NatQuery / NatCDC this section may not be applicable; it will only be applicable if you are using NatQuery / NatCDC to extract data from ADABAS and load this data into a RDBMS residing on a completely separate platform.

To configure Secure FTP against a remote RDBMS platform, perform the following steps:

5. **Start NatQuery**
   Please start NatQuery if it is not already started. If already started, insure that NatQuery is open with an empty NatQuery desktop (I.E. no Query windows or other windows open).

6. **Invoke the Administer Server Connection Information Window**
   On an empty NatQuery desktop, click **Administer** > **Environment Configuration** > **RDBMS Target Configuration** > ***RDBMSName*** (where the "***RDBMSName***" is either "SQL Server", "MySQL" or "Oracle". This action will invoke appropriate **RDBMS Target Configuration – *RDBMSName* - General Defaults** window.

   When this window appears, the first Tab entitled ***RDBMSName* Command Options** (where ***RDBMSName*** is either "SQL Server", "MySQL".

   Click on the **Execution Configuration** Tab to continue.

7. **Configure FTP – Execution Configuration Tab**
   The Execution Configuration Tab provides for the capture of FTP related information.

   a. **Server Type**
      Set the Server Type to the value that represents the type of platform upon which the Target RDBMS resides. Possible values are "Windows" or "UNIX/Linux".

   b. **Transport Mode**
      Set the Transport Mode to be the desired communication Mode. Possible values are "PC Network/Filecopy", "FTP" and "none".

If Secured FTP communication is desired, then this value should be set to “FTP”.

c. **Server Name**
Set the Server Name to be the Universal Resource Locator (URL), Name or IP Address of the platform upon which the RDBMS Server resides.

d. **Remote Execution Enabled**
This checkbox instructs NatQuery as to whether or not NatQuery will attempt to execute the NatQuery-generated processing scripts once they are placed into the target machine.

Further discussion on configuring Remote Execution is described in the NatQuery Installation and Operations Manual.

e. **FTP Information Frame**

v. **Encryption**
This should be set to the type of FTP connection that is desired. Options are “None (Normal FTP)”, “Implicit FTPS” and “Explicit FTPS”; select the type of FTP communication that matches the setting of the FTP Server on the remote RDBMS platform.

vi. **Port**
This should be set to the value of the Port on the remote FTP Server on the RDBMS target that will handle the FTP Connection.

For normal FTP, this is usually “21” and in most cases will be “990” by default when FTPS Implicit or Explicit connections are used.

Set the **Port** setting to the correct value.

vii. **Passive FTP**
If checked, this checkbox will enable Passive (PASV) FTP communication, if left unchecked Passive FTP communication will be disabled.

Usually, **Passive FTP** should be checked.

viii. **Create FTP Logfile**
This checkbox controls whether or not a Log File of FTP Operations is created when a user performs a FTP operation. This Log File can be useful when debugging connections issues, but should be disabled (unchecked) when FTP operations are working properly because a user’s password **IS** recorded in the Log File.

When checked, a log file with the name of:

*userid*_FTP_RDBMS_TRACE.Log

This file is created in the path specified by the NatQuery Environment Path, where "*userid*" is replaced with the User ID of the NatQuery user.

f. **FTP Transport Frame**
The FTP Transport Frame contains a single field; **Execution Directory.**

The path value placed into this text field is relative to the Target RDBMS platform, and represents the directory that automated FTP will make a Change Directory to, and will additionally be used within the NatQuery-generated script so that execution of Load processes may function correctly.

Set this path to the relative path on the target RDBMS platform where FTPed files will be placed (scripts, parameter files, and data) for subsequent loading into the target RDBMS.

g. **Complete RDBMS Target FTP Configuration**
With the above steps completed, Secure FTP against the target RDBMS platform should now be properly configured.

Click the **OK** button to close the **Target Configuration –** ***RDBMSName*** **- General Defaults** window.

8. **Handle Secure FTP Certificate(s)**
With the above steps complete, you can proceed to handle the Secure FTP Certificate(s) needed to complete the Secure FTP configuration against the target RDBMS platform; these are outlined in the section of this manual entitled **Handling Secure FTP Certificates**.

Handling Secure FTP Certificates

When enabling Secure FTP connections, a client machine needs to be given the appropriate Certificate issued by a Certificate Authority (CA) for the platform being accessed. This Certificate is then stored internally in the Windows Registry where it is subsequently automatically accessed when Secure FTP Operations are executed.

If the appropriate Certificates are already installed into the Security Store on the Client Machine then this section may be bypassed.

In the current version of NatQuery, NatQuery itself does not provide any mechanism to Import a Certificate into the Windows Registry as this ability is already inherent in a Windows environment through a function available with Microsoft Internet Explorer and other browsers.

As Internet Explorer (IE) is typically available in every Windows installation, these instructions utilize IE as the mechanism to Import required Certificates.

NOTE: The following process assumes that you have computer access to a digital Certificate created by a Certificate Authority that corresponds to the target FTP platform. If you do not have this Certificate – you will not be able to complete the following steps successfully. Please insure you have path access to the Certificate from the computer you are presently working on.

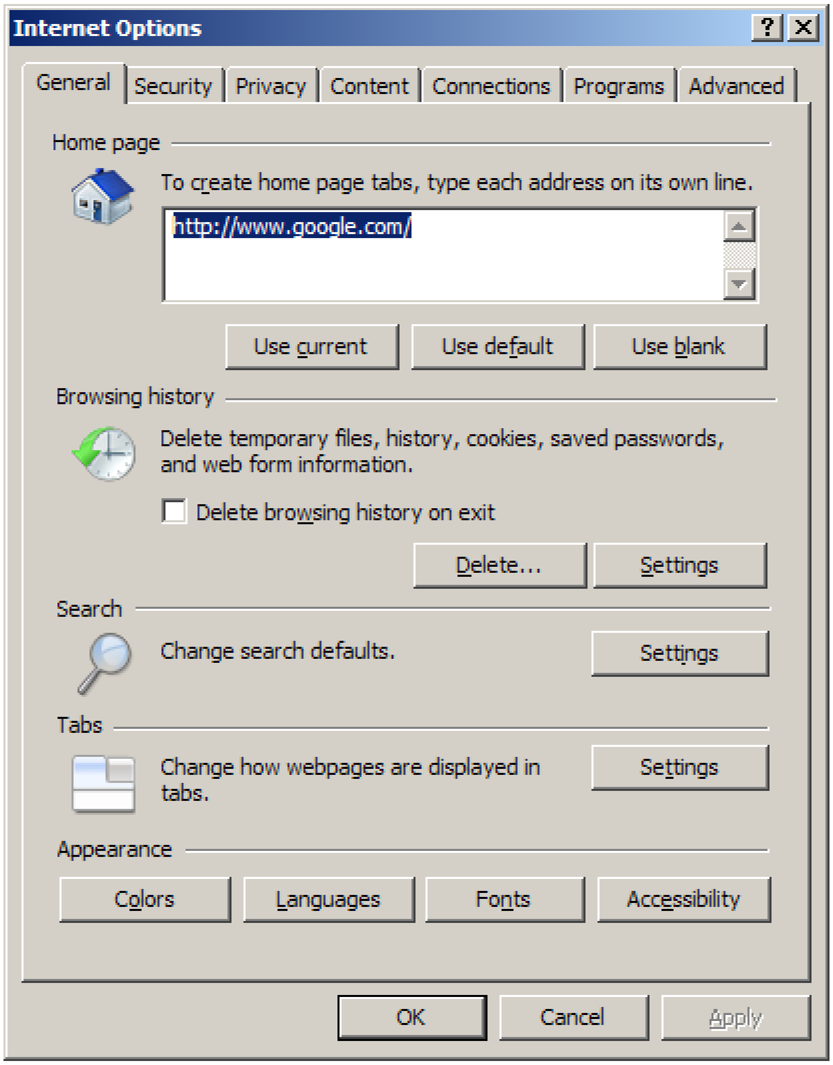
To import a Certificate, perform the following steps:

11. **Start Microsoft Internet Explorer**

    Instructions continue on subsequent pages.

12. Invoke Internet Options

On the IE toolbar will be an item called **Tools**. Clicking on **Tools** will invoke a menu where you will find an **Internet Options** item. Clicking the **Internet Options** item, which will invoke a window similar to the following (the window for IE 8 is shown below):



To create home page tabs, type each address on its own line.

http://www.google.com/

Delete temporary files, history, cookies, saved passwords, and web form information.

□ Delete browsing history on exit

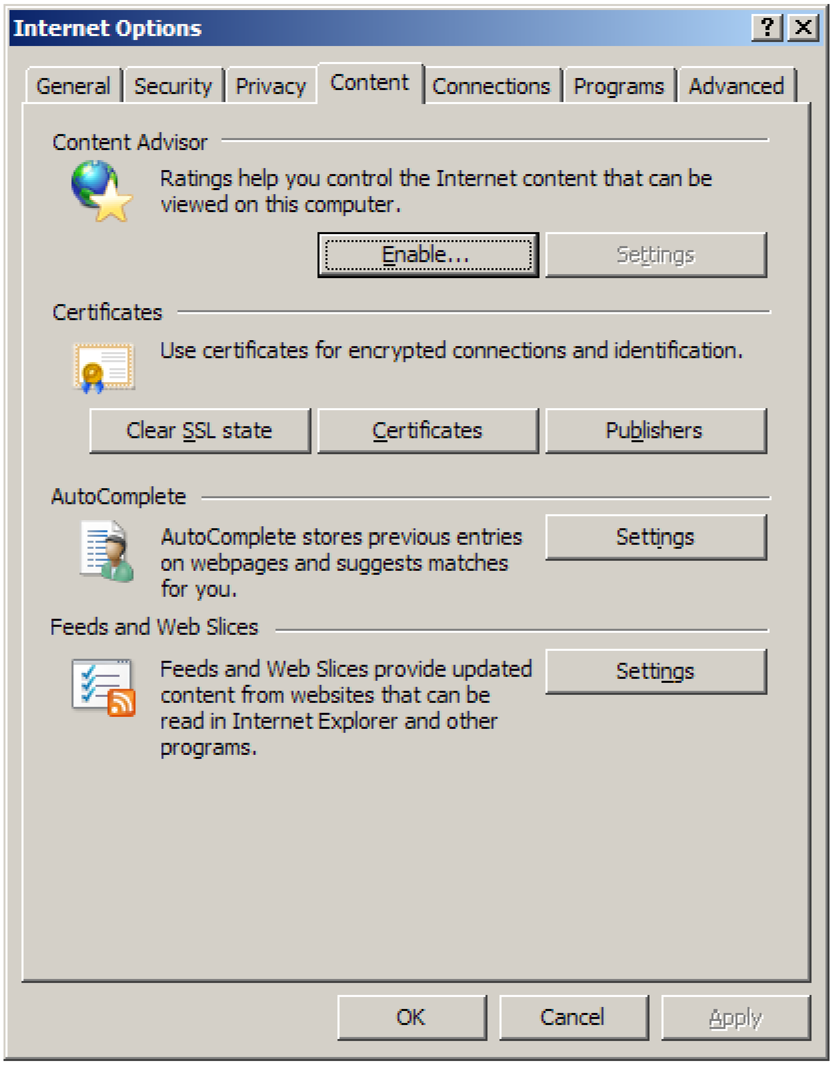
Change search defaults.

Change how webpages are displayed in tabs.

To continue, click on the Tab entitled **Content**.

13. Internet Explorer, Content Tab

Clicking the **Content** Tab as described above will invoke the **Content** Tab, which will look similar to the following:



Content Advisor

Ratings help you control the Internet content that can be viewed on this computer.

Certificates

Use certificates for encrypted connections and identification.

AutoComplete

AutoComplete stores previous entries on webpages and suggests matches for you.

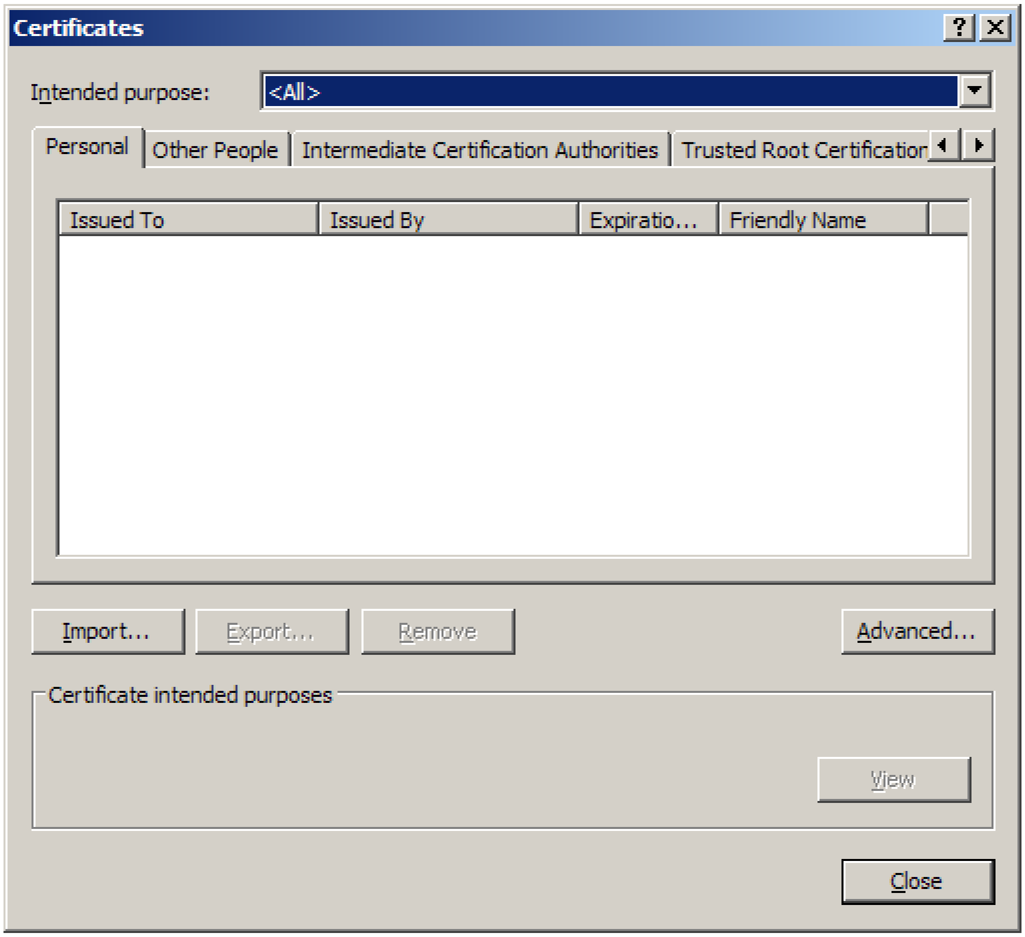
Feeds and Web Slices

Feeds and Web Slices provide updated content from websites that can be read in Internet Explorer and other programs.

To continue, click the **Certificates** button located in the middle of the **Content** Tab. This will invoke a **Certificates** window.

14. Internet Explorer – Certificates Windows

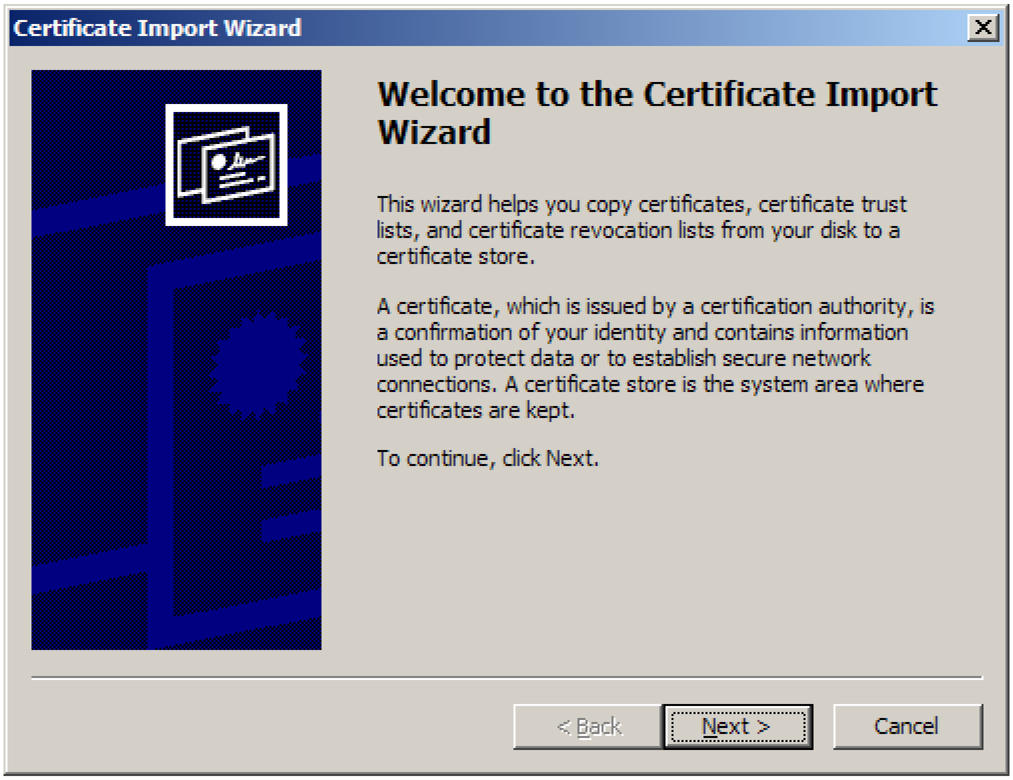
Clicking the Certificates button described above will invoke a window similar to the following:



Intended purpose: <All>

To continue, click the **Import** button on the **Certificates** window.

15. **Internet Explorer – Certificate Import Wizard – Step 1**
    Clicking the **Import** button as described above will invoke the first window of the **Certificate Import Wizard**, which will look similar to the following:



Welcome to the Certificate Import Wizard

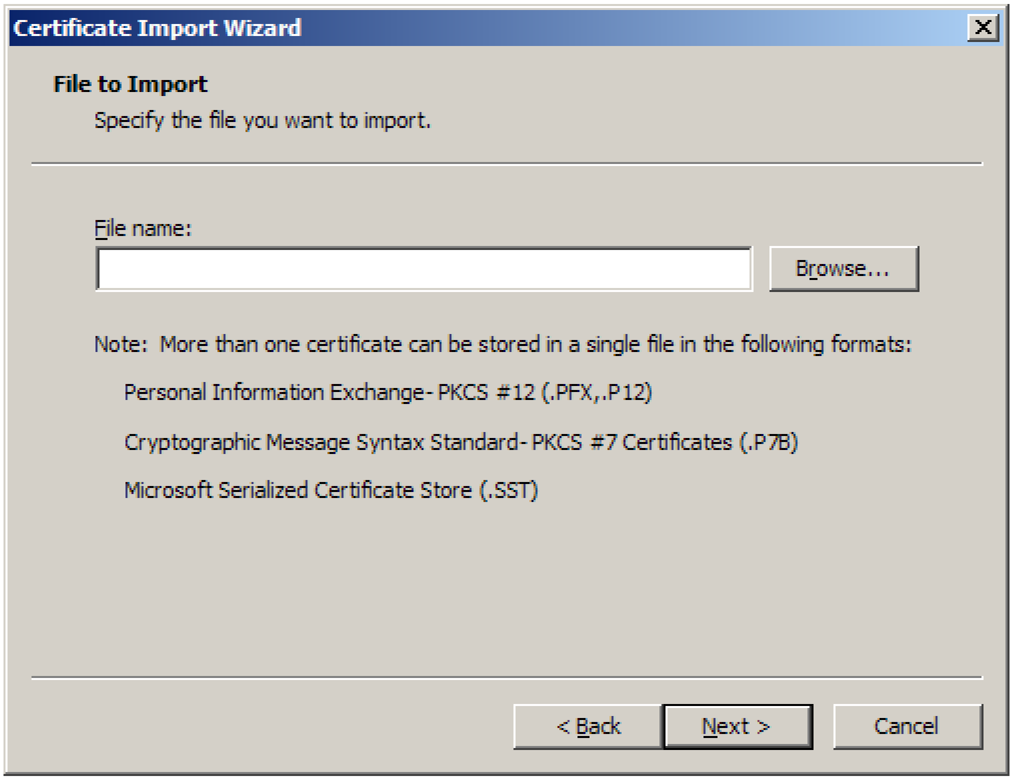
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

To continue, click the **Next** button.

16. **Internet Explorer – Certificate Import Wizard – Step 2**
Clicking the **Next** button as described above will bring up the next screen of the **Certificate Import Wizard**, which will look similar to the following:


Certificate Import Wizard
File to Import
Specify the file you want to import.
File name:
Browse...
< Back
Next >
Cancel

Note: More than one certificate can be stored in a single file in the following formats:

Personal Information Exchange- PKCS #12 (.PFX,.P12)

Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialized Certificate Store (.SST)

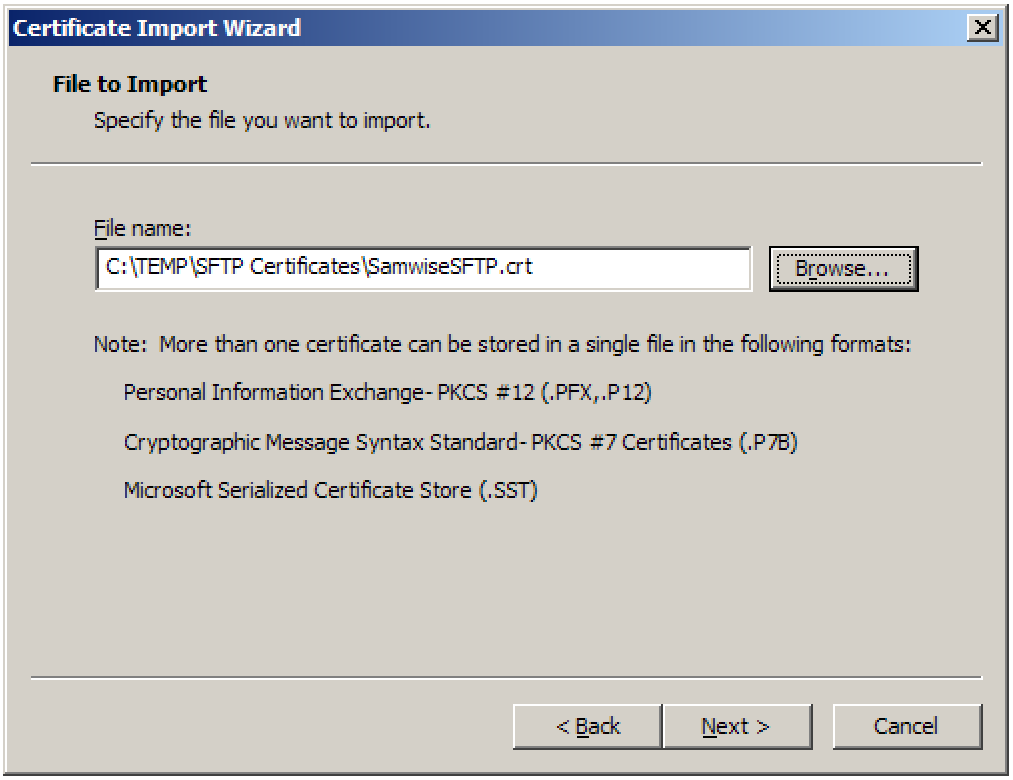
To continue, click the **Browse** button.

17. **Internet Explorer – Certificate Import Wizard – Step 3**
Clicking the **Browse** button as described above will invoke a typical Windows **Open** window that will allow you to navigate to the path where the Certificate that was provided to you has been temporarily stored.

Navigate to the appropriate directory where the Certificate that was given to you has been saved, left-click on it to select / highlight this file.

To continue, then click the **Open** button.

18. **Internet Explorer – Certificate Import Wizard – Step 4**
Performing the above action will return the user to the window seen in #6 above, with the appropriate Certificate file selected. This will now look similar to the following image:


Certificate Import Wizard
File to Import
Specify the file you want to import.
File name:
Browse...
< Back
Next >
Cancel

File name: C:\TEMP\SFTP Certificates\SamwiseSFTP.crt

Note: More than one certificate can be stored in a single file in the following formats:

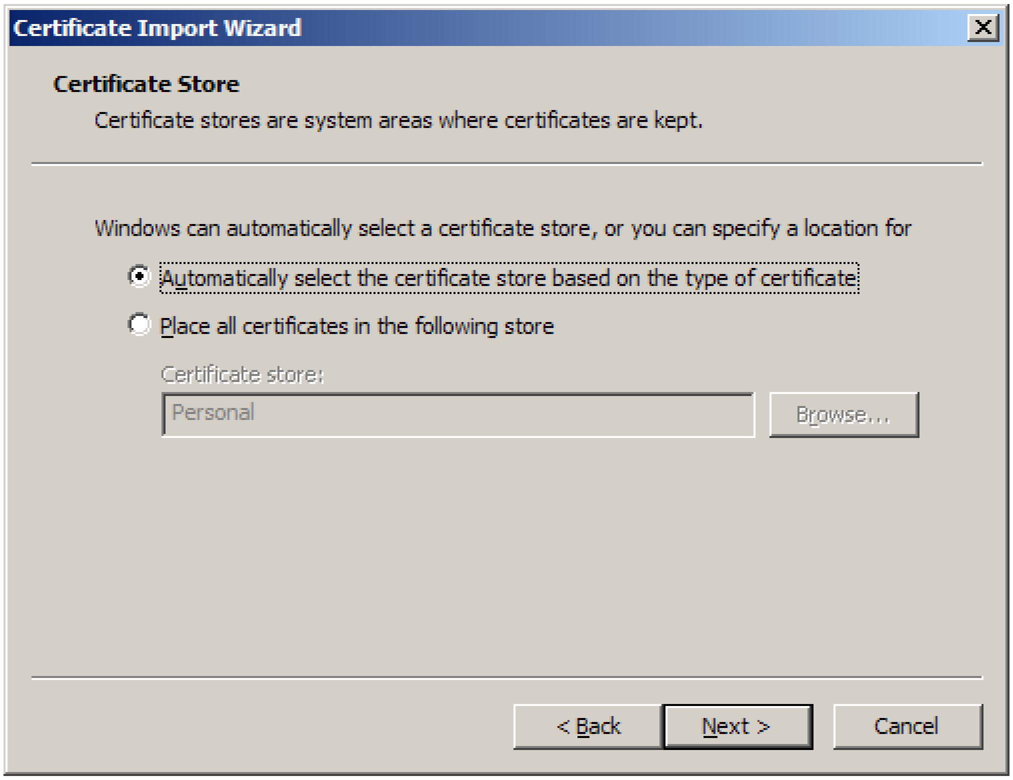
Personal Information Exchange- PKCS #12 (.PFX,.P12)

Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialized Certificate Store (.SST)

To continue, click the **Next** button.

19. **Internet Explorer – Certificate Import Wizard – Step 5**
Performing the above action will invoke a window similar to the following:



Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for

● Automatically select the certificate store based on the type of certificate

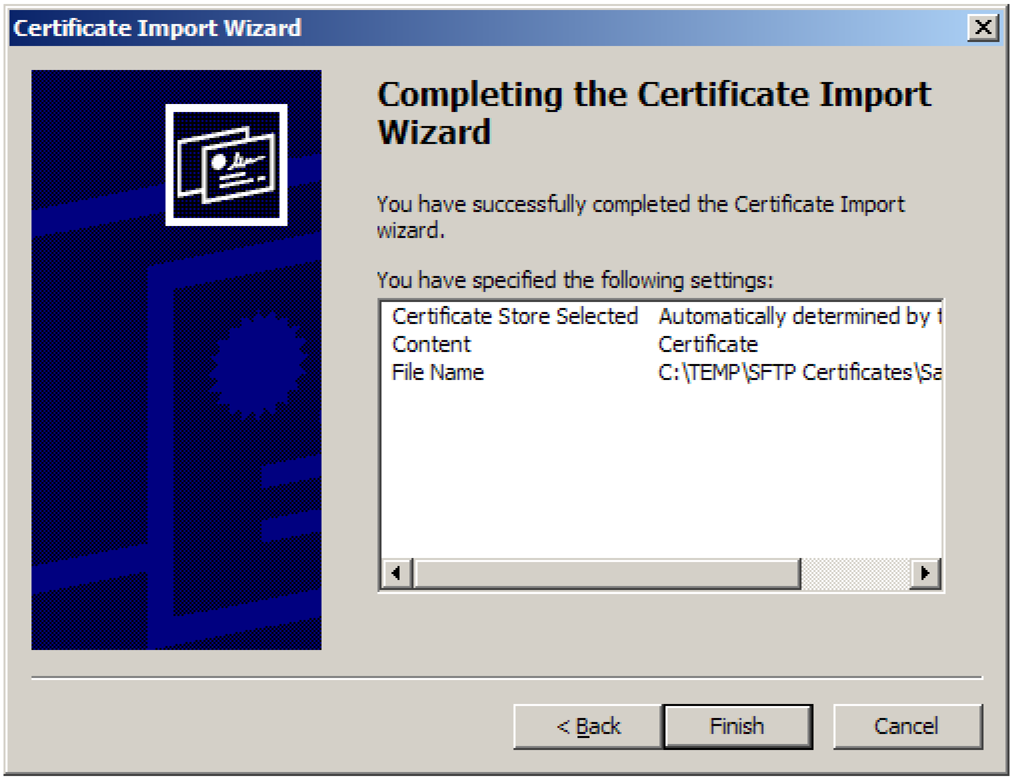
○ Place all certificates in the following store

Certificate store:

Personal

To continue, first click the **Automatically select the certificate store based on the type of certificate** radio button (as shown above), then click the **Next** button.

20. **Internet Explorer – Certificate Import Wizard – Finish**
Performing the above actions should now invoke a window similar to the following:



Completing the Certificate Import Wizard

You have successfully completed the Certificate Import wizard.

You have specified the following settings:

Certificate Store Selected	Automatically determined by t
Content	Certificate
File Name	C:\TEMP\SFTP Certificates\Sa

To continue, click the **Finish** button, after which all remaining open IE windows may be closed as desired.

If the above steps were followed successfully, the appropriate Certificate should now be loaded into the Windows Certificate Store (which is located within the Windows Registry), and Secured FTP communications using NatQuery against the remote FTP platform may occur.

Microsoft's WININET.DLL

The following list summarizes the FTP-related errors that NatQuery may produce, as well as the suggested course of action for each.

- **Un-specified FTP error**
  This error should only occur when problems exist with the operation of the FTP server, or the TCP/IP and or FTP software on the server, or the local network, or the operation of the Windows workstation.

  First retry the FTP operation, and if the problem persists, please contact NatWorks, Inc.

- **2 The FTP local file xxxxxxx does not exist.**
  An FTP PUT operation is being attempted when the file does not physically exist on the workstation in the directory specified.

  By reviewing the FTP log, the file reference that is causing this error should be readily determined. Correct the invalid file reference, and then retry the FTP operation.

  For more information on the FTP Log, please refer to the section entitled

FTP Log File.

- **3 The FTP local directory xxxxxx does not exist.**
  An FTP operation is being attempted that requires a valid reference to a directory on the local workstation, and this directory does not physically exist.

  By reviewing the FTP log, the directory reference that is causing this error should be readily determined. Either correct the invalid directory reference, or create the required directory then retry the FTP operation.

- **123 The FTP remote directory/file name appears to be invalid.**
  An FTP operation is being performed which is referencing a FTP server-based file, and either the "directory" reference and/or the file name is invalid.

  By reviewing the FTP log, the directory reference and/or file name that is causing this error should be readily determined. Using the Administration functions of NatQuery, the directory reference and/or the file name should be examined for correctness in accordance with accepted naming standards on the server, and the FTP operation should then be retried.

- **12001 No more Internet handles could be generated at this time.**
  This error will only occur if the Windows operating system is overloaded with a series of Internet-related operations.

  To resolve this error, it is first suggested that the user wait for outstanding Internet or intranet operations to complete and retry the FTP operation. In extreme situations, this may require that the workstation be rebooted.

- **12002 The FTP request has timed out.**
  This error will occur when an FTP operation "times out" after waiting for a response from the remote system.

  A likely cause of this error is an incorrect IP or URL address for the server as specified to NatQuery through the **Administer / Environment Configuration / Server Connection Configuration / Server Information** function. Verify that the IP address or URL is correct, and then retry the FTP operation.

  Another cause of this problem may involve problems with the server itself.

- **12003 The Log File may be busy and is therefore unavailable**
  This error message is likely to occur when the user is attempting to perform a **Check Server** for Update operation on the **Check Server** window, and one or more of the user's previously submitted requests currently has the user's remote log file open as a result of processing a user request.

  The recommended course of action is to adjust the NATPARM WFOPFA or similar so that the log file is only opened when it is being written to.

Alternatively, the user could wait a bit to allow the currently executing request to finish, and then retry the FTP operation.

- **12003 The FTP remote directory/file name appears to be invalid**
  An FTP operation is being performed which is referencing a server-based file, and either the "directory" reference and/or the file name is invalid.

  By reviewing the FTP log, the directory reference and/or file name that is causing this error should be readily determined. Using the Administration functions of NatQuery, the directory reference and/or the file name should be examined for correctness in accordance with accepted naming standards on the server, and the FTP operation should then be retried.

- **12004 An internal error has occurred with the Internet call.**
  An internal error occurred within the Microsoft DLL that handles Internet operations.

  It is suggested that the user verify that the file named WININET.DLL is present on the workstation, and that the version of this DLL is at least 5.0 or higher.

- **12005 The URL specified appears to be invalid.**
  This error occurs when the structure of the URL or IP address specified to NatQuery for the server does not appear to be in a valid structure.

  A likely cause of this error is an incorrect IP or URL address for the server as specified to NatQuery through the **Administer / Environment Configuration / Server Connection Configuration / Server Information** function. Verify that the IP address or URL is correct, and then retry the FTP operation.

- **12006 The URL scheme could not be recognized or is not supported.**
  This error occurs when the structure of the URL or IP address specified to NatQuery for the server does not appear to be in a valid structure.

  A likely cause of this error is an incorrect IP or URL address for the server as specified to NatQuery through the **Administer / Environment Configuration / Server Connection Configuration / Server Information** function. Verify that the IP address or URL is correct, and then retry the FTP operation.

- **12007 The FTP server name could not be resolved and appears to be invalid.**
  This error occurs when the structure of the URL or IP address specified to NatQuery for the server does not appear to be in a valid structure.

  A likely cause of this error is an incorrect IP or URL address for the server as specified to NatQuery through the **Administer / Environment Configuration / Server Connection Configuration / Server Information** function. Verify that the IP address or URL is correct, and then retry the FTP operation.

- **12008 The requested Internet protocol of FTP could not be located.**
  This error will occur in situations where FTP is not properly configured on the workstation.

  The suggested course of action is to review the installation of the Windows operating system and re-install the required communication component(s) – which would be Microsoft's Internet Explorer.

- **12009 An internally specified Internet Option is not supported.**
  This error will occur when an otherwise valid FTP operation is attempted, but this operation is being rejected by the remote server.

  By reviewing the FTP log, the type of FTP operation being attempted should be readily determined. Depending upon the nature of the operation being requested, one resolution to this problem may be to have the FTP software on the server reconfigured to support the required operation.

  This error will occur in situations where **Direct FTP** operations are being attempted against an MVS mainframe however the FTP software configuration does not allow JES to be used as a target.

- **12010 The length of an internally specified Internet Option is invalid.**
  This error will occur when the internal structure of a FTP operation is invalid.

  It is first suggested that the user verify that the file named WININET.DLL is present on the workstation, and that the version of this DLL is at least 5.0 or higher.

  If this problem persists, please notify NatWorks, Inc.

- **12011 An internally specified option cannot be set, only queried.**
  This error will occur when the internal structure of a FTP operation is invalid.

  It is first suggested that the user verify that the file named WININET.DLL is present on the workstation, and that the version of this DLL is at least 5.0 or higher.

  If this problem persists, please notify NatWorks, Inc.

- **12012 The Win32 Internet function support is being shut down or unloaded.**
  This error occurs when an internal error has occurred within the Microsoft WININET.DLL.

  It is first suggested that the user verify that the file named WININET.DLL is present on the workstation, and that the version of this DLL is at least 5.0 or higher.

If this problem persists, please notify NatWorks, Inc.

- **12013 The supplied FTP User-ID appears to be invalid.**
  This error occurs when an FTP connection is being established and FTP login operation is rejecting the value of the currently supplied User-ID.

  To correct this situation, first ascertain the value of the current User-ID that NatQuery is operating under. This will be seen on the **User Identification** tab of the **NatQuery Configuration** window. Once this **User-ID** is determined, verify that the User-ID is authorized to use FTP operations against the server, and further verify that the spelling and case of this User-ID match what is expected by the server's FTP software.

- **12014 The supplied password appears to be invalid.**
  This error occurs when an FTP connection is being established and FTP login operation is rejecting the value of the user-supplied password that is associated with the currently supplied User-ID.

  To correct this situation, first ascertain the value of the current **User-ID** that NatQuery is operating under. This will be seen on the **User Identification** tab of the **NatQuery Configuration** window. Once this **User-ID** is determined, verify that the password value being entered by the user for the FTP operation matches what is expected by the server in terms of spelling and case.

- **12015 The request to connect / log on to the FTP server failed.**
  This error should only occur when problems exist with the operation of the server, or the TCP/IP and/or FTP software on the server.

  First retry the FTP operation, and if the problem persists, please contact NatWorks, Inc.

- **12016 The requested Internet / FTP operation is invalid.**
  This error should never occur as NatQuery only attempts to use the FTP operations of PUT, GET and optionally DELETE.

  Retry the FTP operation that produced the error. If the problem persists, please contact NatWorks, Inc.

- **12017 The FTP operation was canceled, because the handle was closed.**
  This error will occur in situations where an FTP operation was underway, but the Windows operating system is either in the process of being shutdown, or something affected the Windows operating system in a negative manner.

  If the windows operating system was in the process of being shutdown, this error can be ignored. Otherwise, it is suggested that the user reboot the machine and then retry the FTP operation.

- **12018 The internal Internet handle is incorrect.**
  This error should never occur through NatQuery, if it does, it indicates a severe internal problem within the NatQuery application itself.

  First, retry the FTP operation. If the problem persists, then please contact NatWorks, Inc.

- **12019 The internal Internet handle is not in the correct state.**
  This error should never occur through NatQuery, if it does, it indicates an internal problem within the NatQuery application itself.

  First, retry the FTP operation. If the problem persists, then please contact NatWorks, Inc.

- **12020 The request cannot be made via a proxy.**
  This error indicates that NatQuery was configured to use a Proxy Server to make FTP connections, and the Proxy Server is rejecting the request.

  Proxy Servers can represent a difficult obstacle to overcome when attempting FTP connections and in some cases a Proxy Server will be unable to handle FTP operations at all.

  First, verify that a Proxy Server is required to perform FTP operations between the NatQuery workstation and the remote mainframe.

  If a Proxy Server is required, then please contact NatWorks for information on whether or not NatQuery will be able to support this Proxy Server.

- **12021 A required registry value could not be located.**
  This error is most likely caused by an installation problem with Microsoft Software.

  It is first suggested that the user verify that the file named WININET.DLL is present on the workstation, and that the version of this DLL is at least 5.0 or higher.

  If this problem persists, please notify NatWorks, Inc.

- **12022 A required registry value is incorrect / invalid.**
  This error is most likely caused by an installation problem with Microsoft Software.

  It is first suggested that the user verify that the file named WININET.DLL is present on the workstation, and that the version of this DLL is at least 5.0 or higher.

  If this problem persists, please notify NatWorks, Inc.

- **12023 Direct network access cannot be made at this time.**
  This error is typically caused when a problem exists with the site network.

  Please verify that network software is operating properly, and then retry the FTP operation.

- **12024 An asynchronous request had a zero context value.**
  This error is caused by an invalid internal call for an Internet operation, and should never occur from within NatQuery.

  Please retry the FTP operation, and if the problem persists please contact NatWorks, Inc.

- **12025 An asynchronous request had no callback function set.**
  This error is caused by an invalid internal call for an Internet operation, and should never occur from within NatQuery.

  Please retry the FTP operation, and if the problem persists please contact NatWorks, Inc.

- **12026 One or more requests are already pending.**
  This error is caused by an invalid internal call for an Internet operation, and should never occur from within NatQuery.

  Please retry the FTP operation, and if the problem persists please contact NatWorks, Inc.

- **12027 The format of the request is invalid.**
  This error is caused by an invalid internal call for an Internet operation, and should never occur from within NatQuery.

  Please retry the FTP operation, and if the problem persists please contact NatWorks, Inc.

- **12028 The requested item could not be located.**
  This error is most likely caused by an installation problem with Microsoft Software.

  It is first suggested that the user verify that the file named WININET.DLL is present on the workstation, and that the version of this DLL is at least 5.0 or higher.

  If this problem persists, please notify NatWorks, Inc.

- **12029 The attempt to connect to the server failed. Please verify the Internet / intranet connection.**
  This error should only occur when problems exist with the operation of the server, or the TCP/IP and/or FTP software on the server.

  First retry the FTP operation, and if the problem persists, please contact NatWorks, Inc.

- **12030 The connection with the server has been terminated.**
  This error should only occur when problems exist with the operation of the server, or the TCP/IP and/or FTP software on the server, or the workstation is in the process of being shutdown.

First retry the FTP operation, and if the problem persists, please contact NatWorks, Inc.

- **12031 The connection with the server has been reset.**
  This error should only occur when problems exist with the operation of the server, or the TCP/IP and/or FTP software on the server, or the local network.

  First retry the FTP operation, and if the problem persists, please contact NatWorks, Inc.

- **12032 The requested function should be attempted again.**
  This error should only occur when problems exist with the operation of the server, or the TCP/IP and/or FTP software on the server.

  First retry the FTP operation, and if the problem persists, please contact NatWorks, Inc.

- **12033 The request to the proxy was invalid.**
  This error indicates that NatQuery was configured to use a Proxy Server to make FTP connections, and the Proxy Server is rejecting the request.

  Proxy Servers can represent a difficult obstacle to overcome when attempting FTP connections and in some cases a Proxy Server will be unable to handle FTP operations at all.

  First, verify that a Proxy Server is required to perform FTP operations between the NatQuery workstation and the remote mainframe.

  If a Proxy Server is required, then please contact NatWorks for information on whether or not NatQuery will be able to support this Proxy Server.

- **12036 The request failed: The internal handle already exists.**
  This error should never occur through NatQuery, if it does, it indicates an internal problem within the NatQuery application itself.

  First, retry the FTP operation. If the problem persists, then please contact NatWorks, Inc.

- **12110 The request failed: An FTP operation is already in progress.**
  This error message should only occur when an FTP operation is already underway and a second FTP operation is attempted.

  This error should never occur through NatQuery, if it does, it indicates an internal problem within the NatQuery application itself.

  First, retry the FTP operation. If the problem persists, then please contact NatWorks, Inc.

- **12111 The request failed: The FTP session was aborted.**
  This error should only occur when problems exist with the operation of the server, or the TCP/IP and/or FTP software on the server, or the local network.

First retry the FTP operation, and if the problem persists, please contact NatWorks, Inc.

Job Control Language (JCL) / Execution Issues

The physical execution of a given NatQuery request is designed to execute in a server's "batch" environment. To support the automated execution of a NatQuery request, the Administrator builds Job Control Language (JCL) templates or Script templates that NatQuery then uses to support this "batch" execution.

When NatQuery is first being configured, it is highly likely that problems will be encountered when these JCL / Script templates are initially executed. Depending upon the severity of the problems encountered, NatQuery executing on a workstation may be unable to determine that problems exist with requests submitted for execution. This is due to the fact that NatQuery depends upon these JCL / Script streams to "log" their actions into server-based files designated as User Log Files. If JCL / Script problems prevent these log files from being properly updated, then NatQuery has no way of determining that a problem was encountered.

Particularly for initial installation and configuration of NatQuery when various JCL / Script processes are first being executed, it is strongly suggested that the Administrator monitor the execution of submitted JCL / Script as well as the execution results of submitted JCL / Script so that problems with JCL / Script execution are immediately discovered. Once various JCL / Script templates are properly configured, further monitoring should not be needed.

Simplistically, JCL / Script templates are generic to a given task. Those elements in a JCL / Script template that will make a generic template specific to a given task are designated as "dynamic variables", with these variables being identified by a proceeding "&&". As a final step in the submission of a given request, NatQuery selects the appropriate JCL / Script template that will support the required task, and then performs substitutions against these pre-designated dynamic variables, replacing these dynamic variables with appropriate values. In this manner, generic JCL / Script templates are transformed into specific JCL / Script templates.

When problems arise with the execution of JCL / Script streams that are handled by NatQuery, the Administrator should first determine whether the cause of the problem is related to the structure and layout of the specific JCL / Script template involved, or whether the problem is attributable to the dynamic substitutions made by NatQuery.

Other topics of interest relating to JCL / Script would be:

- Extract Type to Template Reference,
- Diagnosing JCL / Script Problems, and
- Problems with Dynamic Substitutions into JCL /Script Templates.

Extract Type to Template Reference

The following table outlines the NatQuery Extract Types that utilize JCL / Script templates, and relates these Extract Types to the template used by the Extract Type.

NatQuery Extract Type / Function	Template Utilized
Download to PC File Download into Excel SourcePoint Extract Download into Access RDBMS Loading Download to XML File	Production Request Process (default) Production Sequential Process (if seq file) Production Summary Process (opt.)
Data Discovery Functions Descriptor Statistics	Production Request Process
Download to PC File	Production E-Mail Process (opt.)
DWH Software Extract	Production DWH Process
Extract to XML on Server	Production XMLServer Process
Request Predict Information	Production Predict Process
DBS Loader	Production DB2 Loader
ADACMP Decompress INFILE	Production ADACMP Process
ADAULD Extract	Production ADAULD Process
ADAULD Extract	Production ADAULD Savetape
Download DDM	Production Special Process
Download FDT	Production Special Process
Upload NatQuery Server Programs	Production Request Process
Process Data Into ADABAS	Production ADABAS Process

Diagnosing JCL / Script Problems

In order to effectively diagnose JCL / Script execution problems, it is usually required that the Administrator must examine the output of a failed request. By examining this output, the cause of the failure can usually be easily determined and then corrected.

When NatQuery creates a request, this request is generally designed to invoke Natural in a batch environment. The output of this batch execution is therefore controlled by the setting of CMPRINT, with the setting of CMPRINT occurring in the batch JCL / Script template itself.

In most cases, a failed execution typically resolves itself to be a problem in the JCL / Script itself, usually a syntax error. In other cases, files may be incorrectly referenced or even miss-spelled, but in all cases the problem is usually quickly identified by looking at the CMPRINT output file from a failed NatQuery execution.

By reviewing the out list, it is usually seen that changes are required in a corresponding JCL / Script template. These changes should then be made, and the function that failed should be re-tried.

Changes to JCL / Script templates are made through the **Administer JCL / Script** function. This function is accessible to a NatQuery installation running in Administrator mode by clicking on the **Administer** drop-down menu, then clicking on **Environment Configuration**, then clicking on **Server Connection Configuration**, and then clicking on **JCL/ Script Information**.

Once on the **Administer JCL / Script** window, the Administrator would select the JCL / Script template that is causing the problem. Using the editor capabilities of the **Administer JCL / Script** window, the Administrator will then make appropriate changes to the specific JCL / Script template, will **Save** these changes, then close the **Administer JCL / Script** window, re-verify the **Environment Configuration** and should then re-try the previously failed request.

Problems with Dynamic Substitutions into JCL /Script Templates

If the cause of the JCL /Script execution problem is related to the dynamic substitutions that NatQuery makes into any given template, then the Administrator must determine which dynamic substitutions are in error and then use one or more of the various administrative functions to modify the incorrect value that is being substituted.

It should be noted that in versions of NatQuery prior to 2.4.2, dynamic substitution variables were designated in all lower case. All later versions expect dynamic-substitution variables to be in upper case. If you have upgraded from a 2.4.1 version to 2.4.2 and are experiencing dynamic substitution problems, then all JCL / Script templates should be examined to insure the use of upper-case substitution values.

To correct dynamic substitution problems, the Administrator must first determine which dynamic substitution variable is failing, and then determine the proper corrective action to insure the substitution of proper information. To assist the Administrator in determining where each NatQuery Dynamic substitution variable is initialized, the Administrator should refer to the section entitled NatQuery Dynamic Substitution Variable Reference Table.

Once the Administrator has determined which function controls the value being dynamically substituted by NatQuery, the Administrator should use the appropriate function to correct the incorrect value. Once corrections are made, the Administrator should re-verify the current Environment Configuration function and then retry the request submission.

Documentation & Help

This NatQuery Installation & Operations Guide contains all information required to install NatQuery and perform the basic configuration tasks.

Additional documentation on features and aspects of NatQuery can be found in the workstation-based Help system. This Help system is available immediately after installation and once the NatQuery application has been started.

To access the Help system from the NatQuery desktop, the user should click the Help menu drop-down while in NatQuery.

To access window-specific Help, the user may either click the F1 key while on the window of interest, or can achieve the same results by clicking the Help button that can be found in the lower right of every window that NatQuery displays.

Contacting NatWorks

NatWorks realizes that the person(s) who will install, administer, and use NatQuery for ADABAS data extraction may require assistance. We are pleased to provide the following methods of contacting us.

Please note that if you organization purchased NatWorks products through a partner company of NatWorks, you will usually use that partner as you primary source of contact such that your communication with NatWorks flows through this partner company. While NatWorks will never reject any call for assistance, it is desirable that communication be handled in this manner – and in some cases this will be contractually required.

The preferred methods of contacting NatWorks would be:

- NatWorks Website,
- Phone Support, and
- E-Mail Support

NatWorks Website

When looking for general information or to download the latest versions of NatQuery, the reader should refer to the NatWorks, Inc. website located at URL http://www.natworks-inc.com.

Phone Support

If you are experiencing a problem with NatQuery, or have questions or concerns that require immediate attention, you may call NatWorks during normal business hours from 8:30 AM to 5:30 PM, Eastern Standard Time (EST), at (802) 485-6112.

E-Mail Support

If you are experiencing a problem with NatQuery, or have questions or concerns that do not require immediate attention, then you can contact NatWorks, Inc. through e-mail using the following addresses:

- Technical Support support@natworks-inc.com
- General Information info@natworks-inc.com