IBM Net.Data for OS/400

Net.Data Return Codes

Version 5

IBM Net.Data for OS/400

Net.Data Return Codes

Version 5

Note

Before using this information and the product it supports, read the information in "Notices" on page 9.

June 2001 Edition

This edition applies to:

- IBM HTTP Server for AS/400® (Program 5769-DG1), Version 4 Release 4 Modification 0
- IBM HTTP Server for iSeries® (Program 5722-DG1), Version 5 Release 1 Modification 0

and to all subsequent versions and releases until otherwise indicated in new editions.

Preface

Thank you for selecting Net.Data®, the IBM® development tool for creating dynamic Web pages! With Net.Data you can rapidly develop Web pages with dynamic content by incorporating data from a variety of data sources and by using the power of programming languages you already know.

About this book

This book describes return codes and error messages returned by Net.Data.

This book might refer to products or features that are announced, but not yet available.

More information including sample Net.Data macros, demos, and the latest copy of this book, is available from the following World Wide Web site: http://www.ibm.com/systems/i/software/netdata.

Who should read this book

People who want to understand Net.Data error codes and messages returned by Net.Data.

To understand the concepts discussed in this book, you should be familiar with the information in the *Net.Data Administration and Programming Guide* and *Net.Data Reference*.

Contents

Return codes

This section describes the codes returned from Net.Data functions. A return code can be positive or negative. A negative return code indicates an internal Net.Data error, which a positive return code indicates an error within the function, such as syntax, invalid values, or a system error.

-1002

Explanation: A call to an internal function failed. This is a Net.Data internal error.

User response: Report the problem to your software service representative.

-1001

Explanation: The server could not process a Net.Data request to allocate memory.

User response: Ensure the server has enough memory available.

1000

Explanation: The function requested on a function call is not a supported Net.Data built-in function.

User response: Ensure the specified call to the built-in function doesn't have any typographical errors. Refer to the *Net.Data Reference* for a list of the supported built-in functions.

1001

Explanation: An input parameter contained a NULL value.

User response: Ensure the parameters are defined and not NULL before they are passed to the function.

1002

Explanation: An input parameter contained a string value which consisted of the null-terminating character.

User response: Ensure the specified parameter does not contain a NULL value.

1003

Explanation: An incorrect number of parameters were passed on a function call.

User response: Check the function syntax and ensure you pass all the required parameters and no more than the maximum allowed.

1004

Explanation: A parameter passed on a function call, required to be a Net.Data macro table variable, was of a different variable type.

User response: Ensure the variable is defined as a TABLE variable in a DEFINE statement or block.

1005

Explanation: A parameter passed on a function call, required to be a string variable, was of a different variable type.

User response: Ensure the variable is defined as a string variable in a DEFINE statement or block.

1006

Explanation: A literal string was passed on a function call for a parameter which was required to be an output parameter.

User response: Do not specify any input values for output parameters. The parameter type might need to be changed to INOUT.

1007

Explanation:

A parameter contains a value which is not valid. One of these conditions exists:

- A value was passed which exceeded the maximum supported value.
- A value was passed which was less than the minimum supported value.
- A value was passed which was not one of the supported choices.
- A table row or column value was passed which was less than or equal to zero.

User response: Ensure that the value is not out of range or invalid.

1008

Explanation:

A parameter is outside of table bounds. One of these conditions exists:

- A program attempted to modify a table's row or column value, but the row or column value received was less than 0 or greater than the maximum number of rows allowed in the table.
- A row or column value was received as input to a built-in function, but the value received was less than 0 or greater than the current number of rows or columns in the table.

User response: Ensure the specified value is not less than 0 or greater than the current number of rows in the table.

1009

Explanation:

The syntax of the data returned by a System or Perl program is not in the correct Net.Data string format. One of these conditions exists:

- An equal sign was not found.
- A beginning quote was not found.
- An ending quote was not found.
- A space separator between values was not found.

User response: Check the data returned by the function for syntax errors and modify the script accordingly.

1010

Explanation: Not all of the requested data could be returned. A table was specified as an output parameter, but the number of rows of data returned by the language environment was greater than the maximum number of rows allowed for the table. Data was written to the table until it was full, and the remainder of the data was discarded.

User response: You can ignore the discarded data, or increase the table size and run the function again.

2000

Explanation: A flat file interface built-in function could not find the specified file.

User response: Ensure the file is in a path specified by the FFI_PATH statement in the initialization file.

2001

Explanation: A flat file interface built-in function could not open the specified file because it was in use by this or another process, and could not be shared in the specified mode.

User response: Ensure another process has not locked the file.

2002

Explanation: A flat file interface built-in function could not close the specified file because it was not opened by this macro invocation.

User response: The file must be closed by the macro that opened it. Changes made could be lost.

2003

Explanation: A flat file interface built-in function could not read a row of data into a table variable because the number of bytes in the row exceeded the maximum supported number of bytes.

User response: The table is too large for Net.Data to handle.

2004

Explanation: A flat file interface built-in function was attempting to find a file, but encountered a path in the FFI_PATH configuration file variable that was longer than the maximum supported number of bytes, which is 4095.

User response: Shorten the FFI_PATH statement to only those directories Net.Data needs for the current application.

2005

Explanation: A call to a system function failed. This is an internal error reported to Net.Data that may require user interaction or it may be a temporary system error that is not appropriate for Net.Data to handle. If this problem persists, report the problem to your software service representative.

User response: Check your configuration and try again. If problems persist, report the problem to your software service representative.

2006

Explanation: A flat file interface built-in function could not access the specified file because it was in use by this or another process and could not be shared in the specified mode.

User response: End the process using the file and try again. Consider specifying a RETRY value to automatically retry if the file is in use when the function is called.

3001

Explanation: A Web registry built-in function could not create a Web registry because the specified registry already exists.

User response: Use another name for the Web registry.

3002

Explanation:

A Web registry built-in function could not delete the specified registry. One of these conditions exists:

- The registry was in use by another process.
- The registry could not be found.

User response: If the registry is in use by another process, try after the process has closed the registry.

3003

Explanation: A Web registry built-in function could not add an entry to the specified registry because the specified entry already exists.

User response: Duplicate entries cannot be made in a Web registry. Modify the entry and resubmit the function, or use the existing entry.

3004

Explanation: A Web registry built-in function could not remove or retrieve an entry from the specified registry because the specified entry does not exist.

User response: Net.Data cannot find the requested registry entry.

3005

Explanation: A Web registry built-in function could not use the specified registry because it cannot be found.

User response: Create the registry if it does not already exist.

3006

Explanation: A Web registry built-in function could not create the specified registry because a path in the registry name does not exist.

User response: Specify a valid path when you create the registry.

3007

Explanation: A Web registry built-in function could not complete the specified operation because the requestor does not have the proper authority to the specified registry.

User response: Change the security parameter in your Web registry functions to authorize the operation.

3008

Explanation: A Web registry built-in function could not create the specified registry for unknown reasons.

User response: Check your configuration and try again.

4000

Explanation:

A parameter contains an invalid whole number value. One of these conditions exists:

- An input parameter contained a value which was not a whole number.
- An input parameter contained a value which was greater than the supported maximum of 999,999,999.
- An output cannot be expressed as a whole number.

User response: Ensure that the value is not out of range or invalid.

4001

Explanation:

A parameter contains an invalid number value. One of these conditions exists:

- An input parameter contained a value which was not a valid format for a number.
- An input parameter contained a value which specified an exponent outside the supported range of -999,999,999 to +999,999,999.

User response: Ensure that the value is not out of range or invalid.

4002

Explanation: The result of an arithmetic operation had an exponent that was outside the supported range of -999,999,999 to +999,999,999.

User response: Ensure that the value is within the supported range.

5000

Explanation: The string specified in the EXEC statement of a function block contained only space characters.

User response: Specify a string that is not all space characters.

6000

Explanation: A EXEC statement was not specified in the function block for the function being called.

User response: Add the EXEC statement in the FUNCTION block.

6100

Explanation: The path in EXEC block not valid or the EXEC block is missing.

User response: Ensure that an EXEC block is specified and that the path specified in the EXEC block references a program object.

6101

Explanation: Incorrect attributes for the data type were specified or Net.Data does not support the data type.

User response: Correct the parameter that is in error by using one of the following methods:

- If precision and scale were specified, ensure that scale is less or equal to precision.
- If precision or scale was specified, ensure that the precision or scale is allowed for the data type and that the specified value is within the allowed range.
- If data type is DTWTABLE, ensure that the parameter value is a Net.Data table.
- Ensure that the specified data type is one that is supported.

6102

Explanation: A data conversion error occurred when data was converted from displayable format to internal format or from internal format to displayable format.

User response: Ensure that the data specified is compatible with the data type of the parameter.

6202

Explanation: Java application ended due to an exit call.

User response: Ensure that Java application does not issued exit call.

6203

Explanation: Java process received a terminating signal that was not caught.

User response: Determine why the terminating signal was issued and ensure the signal is not issued.

6204

Explanation: Java process was ended.

User response: Determine why the Java process ended and ensure the process is not ended.

6205

Explanation: Java process ended because of an exception.

User response: Determine why the exception occurred and change the Java application so that the exception does not happen.

7000

Explanation: Net.Data is unable to connect to the specified SMTP server.

User response: Make sure that you have specified the correct hostname or IP number. An SMTP server must be running on port 25 on the specified host.

7001

Explanation: An SMTP error occurred while Net.Data tried to relay the e-mail message to the specified SMTP server.

User response: Make sure that you call DTW_SENDMAIL with valid parameters.

7002

Explanation: The specified SMTP server does not support the Extended Simple Mail Transfer Protocol (ESMTP). Net.Data cannot relay e-mail messages that contain 8-bit characters (such as non-U.S. ASCII characters) to an SMTP server that supports only 7-bit characters (such as U.S. ASCII).

User response: Either specify an SMTP server that supports ESMTP (8BITMIME) or do not use 8-bit characters in your e-mail messages.

8000

Explanation:

The cookie cannot be found. One of the following conditions exists:

- The cookie has never been set.
- The cookie has expired.
- The cookie does not have an expiration date and is not persistent; the Web browser that received the cookie exited or was ended.
- The cookie was set with a secure option, and the current HTTP request was sent over an insecure channel.
- The Web browser did not accept cookies or it did not execute JavaScript programs at the time when the set cookie request was submitted.
- The cookie has been deleted by the Web browser.

User response: Determine the reason why the cookie cannot be found and correct the situation.

8200

Explanation: Macro persistence is not enabled. A call was made to a built-in function that requires macro persistence to be enabled.

User response: Enable persistance within the macro or remove the function call. Refer to the *Net.Data Administration and Programming Guide* for information on enabling persistent macros.

8201

Explanation: A persistent built-in function was called out of sequence.

User response: Ensure that the persistent built-in functions are correct and in sequence. For example, you cannot have a call to DTW_ACCEPT() followed by a call to DTW_TERMINATE(). To learn about the proper sequence of persistent built-in functions, see the persistent macro chapter in the *Net.Data Administration and Programming Guide*.

8202

Explanation: Persistence could not be enabled. The attempt to define a static variable failed because text prior to the variable has been written to stdout.

User response: Ensure that persistence is enabled by defining the static variable before writing text to stdout.

Appendix. Net.Data technical library

The Net.Data technical library is available from the Net.Data Web site at http://www.ibm.com/systems/i/software/netdata/docs.

Document	Description
Net.Data Administration and Programming Guide	Contains conceptual and task information about installing, configuring, and invoking Net.Data. Also describes how to write Net.Data macros, use Net.Data performance techniques, use Net.Data language environments, manage connections, and use Net.Data logging and traces for trouble shooting and performance tuning.
Net.Data Reference	Describes the Net.Data macro language, variables, and built-in functions.
Net.Data Language Environment Interface Reference	Describes the Net.Data language environment interface.
Net.Data Messages and Return Codes	Lists Net.Data error messages and return codes.

Related documentation

The following documents might be useful when using Net.Data and related products:

- *DB2 for IBM i SQL Programming*
- *IBM i Distributed Database Programming*

Additionally, OS/400 documentation and redbooks, including books about DB2, are available at the following URL:

`http://publib.boulder.ibm.com/iseries/`

Notices

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

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

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

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

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

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
AS/400
DB2
DB2 Universal Database
DRDA
DataJoiner
IBM
IMS
Language Environment
MVS/ESA
Net.Data
OS/2
OS/390
OS/400
OpenEdition

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

Glossary

absolute path
The full path name of an object. Absolute path names begins at the highest level, or "root" directory (which is identified by the forward slash (/) or back slash (\) character).

ANSI American National Standard for Information Systems

API Application Programming Interface

applet A Java program included in an HTML page. Applets work with Java-enabled browsers, such as Netscape Navigator, and are loaded when the HTML page is processed.

BLOB Binary large object.

CGI Common Gateway Interface.

CLOB Character large object.

commitment control
The establishment of a boundary within the process that Net.Data is running under where operations on resources are part of a unit of work.

Common Gateway Interface (CGI)
A standardized way for a Web server to pass control to an application program and receive data back.

cookie A packet of information sent by an HTTP server to a Web browser and then sent back by the browser each time it accesses that server. Cookies can contain any arbitrary information the server chooses and are used to maintain state between otherwise stateless HTTP transactions. *Free Online Dictionary of Computing*

current working directory
The default directory of a process from which all relative path names are resolved.

database
A collection of tables, or a collection of table spaces and index spaces.

database management system (DBMS)
A software system that controls the creation, organization, and modification of a database and access to the data stored within it.

DATALINK
A DB2® data type that enables logical references from the database to a file stored outside the database.

data type
An attribute of columns and literals.

DBCLOB
Double-byte character large object.

DBMS
Database management system.

firewall
A computer with software that guards an internal network from unauthorized external access.

flat file interface
A set of Net.Data built-in functions that let you read and write data from plain-text files.

HTTP HyperText Transfer Protocol

hypertext markup language
A tag language used to write Web documents.

hypertext transfer protocol
The communication protocol used between a Web server and browser.

Internet
An international public TCP/IP computer network.

Intranet
A TCP/IP network inside a company firewall.

Java An operating system-independent object-oriented programming language especially useful for Internet applications.

language environment
A module that provides access from a Net.Data macro to an external data source such as DB2 or a programming language such as REXX.

LOB Large object.

middleware
Software that mediates between an application program and a network. It manages the interaction between a client application program and a server through the network.

null A special value that indicates the absence of information.

path A search route used to locate files.

path name
Tells the system how to locate an object. The path name is expressed as a sequence of directory names followed by the name of the object. Individual directories and the object name are separated by a forward slash (/) or back slash (\) character.

persistence
The state of keeping an assigned value for an entire transaction, where a transaction spans multiple Net.Data invocations. Only variables can be persistent. In addition, operations on resources affected by commitment control are kept active until an explicit commit or rollback is done, or when the transaction completes.

port A 16-bit number used to communicate between TCP/IP and a higher level protocol or application.

registry
A repository where strings can be stored and retrieved.

relative path name
A path name that does not begin at the highest level, or "root" directory. The system assumes that the path name begins at the process's current working directory.

secure endpoint URL
Endpoint beginning with https

SSL Secure Sockets Layer

TCPIP Transmission Control Protocol/Internet Protocol

uniform resource locator
An address that names a HTTP server and optionally a directory and file name, for example: `http://www.ibm.com/software/data/net.data/index.html`.

unit of work
A recoverable sequence of operations that are treated as one atomic operation. All operations within the unit of work can be completed (commited) or undone (rolled back) as if the operations are a single operation. Only operations on resources that are affected by commitment control can be committed or rolled back.

URL Uniform resource locator.

Web server
A computer running HTTP server software, such as Internet Connection.

wire All the underlying components that are responsible for physically sending or receiving a message on the web

XML eXtensible Mark-up Language

Index

Printed in USA