

Messages
Version 4.3
October 1999

CONSTRUCT SPECTRUM™

Manual Order Number: **SPV430-060IBW**

Copyright © SAGA SOFTWARE, Inc., 1999. All rights reserved.

SAGA, SAGA SOFTWARE, the SAGA logo, Free Your Information, the FYI logo, CRIS, Construct, Construct Spectrum, Construct Spectrum SDK, iXpress, Sagacertify, Sagagallery, Sagavista, and Your Fastest Route to Enterprise Integration are trademarks or registered trademarks of SAGA SOFTWARE, Inc. in the U.S. and/or other countries. Adabas, Adabas Delta Save Facility, Adabas Fastpath, Adabas SQL Server, Adabas Vista, Adaplex+, Bolero, Com-plete, Entire, Entire Access, Entire Net-work, EntireX, EntireX DCOM, Entire Broker, Entire Broker SDK, Entire Broker APPC, Entire SAF Gateway, Natural, Natural Architecture, Natural Elite, Natural New Dimension, Natural Lightstorm, Natural Vision, New Dimension, PAC, Predict, Software AG, and Super Natural are developed by Software AG of Darmstadt, Germany and are distributed in the U.S., Latin America, Canada, Israel and Japan exclusively through SAGA SOFTWARE, Inc. and its subsidiaries and distributors. Adabas and Natural are registered trademarks of Software AG of Darmstadt, Germany. Except for Adabas and Natural, these products developed by Software AG of Darmstadt, Germany are either registered trademarks or trademarks of SAGA SOFTWARE, Inc., in the U.S. and/or other countries.

Other company or product names mentioned are used for informational purposes only and may be trademarks or servicemarks of their respective owners.

TABLE OF CONTENTS

PREFACE

How this Manual is Organized	6
Conventions Used in this Manual	7
Related Documentation	9
Construct Spectrum SDK	9
Construct Spectrum	9
Natural Construct	10

1. RUNTIME MESSAGES

Introduction	12
Resolving the Error	13
Checking Library Image File Definitions	13
Messages from the Spectrum Dispatch Client	15
Messages from the Framework Classes	23

2. COMMUNICATION MESSAGES

Introduction	28
Resolving the Error	30
Dispatcher Object Error Properties	30
DisplayErrors Property	30
ErrorMessage Property	31
ErrorNumber Property	31
ErrorSource Property	31
ErrorValue Property	32
RequestProperty Property	32
Successful Property	33
How Entire Broker Calls Are Used	33
Communication Messages	36
Entire Broker Messages	36
Construct Spectrum Messages	38

3. SPECTRUM SYSTEM MESSAGES

Introduction	46
Entire Broker Messages	47
Error Class 0007	47
Error Class 0020	48
Error Class 0021	49
Error Class 0074	52
Error Class 0079	52
Error Class 0215	53
Construct Spectrum Messages	54
License Key Messages	85
Subprogram Proxy Messages	87
Natural Messages	89

PREFACE

Construct Spectrum Messages is for application developers and administrators who want to quickly find information about Construct Spectrum programming messages.

The following topics are covered:

- **How this Manual is Organized**, page 6
- **Conventions Used in this Manual**, page 7
- **Related Documentation**, page 9

How this Manual is Organized

Construct Spectrum Messages describes the error messages you may receive when developing and running Construct Spectrum applications. Construct Spectrum returns three types of error messages:

- Runtime
- Communication
- Construct Spectrum system

Runtime and communication messages are returned to the Spectrum dispatch client. To view the Spectrum system error messages, you must access the Construct Spectrum Administration subsystem.

The chapters in this manual are organized according to message type. Each chapter contains general troubleshooting information, followed by the messages arranged according to source (where appropriate) and message number.

This manual is divided into the following chapters:

Chapter Title	#	Description
Runtime Messages , page 11	1	Contains troubleshooting information and messages returned for runtime errors.
Communication Messages , page 27	2	Contains troubleshooting information and messages returned for communication errors.
Spectrum System Messages , page 45	3	Contains troubleshooting information and messages returned for Spectrum system errors.

Conventions Used in this Manual

This guide uses the following typographical conventions

Example	Description
Introduction	Bold text in cross references indicates chapter and section titles.
“A”	Quotation marks indicate values you must enter.
Browse model, GotFocus, Enter	Mixed case text indicates: <ul style="list-style-type: none"> • The names of Natural Construct and Construct Spectrum editors, fields, files, functions, models, panels, parameters, subsystems, variables, and dialogs. • The names of Visual Basic classes, constants, controls, dialogs, events, files, menus, methods, properties, and variables. • The names of keys.
Alt+F1	A plus sign (+) between two key names indicates that you must press the keys together to invoke a function. For example, Ctrl+S means hold down the Ctrl key while pressing the S key.
CHANGE-HISTORY	Uppercase text indicates the names of Natural command keywords, command operands, data areas, help routines, libraries, members, parameters, programs, statements, subprograms, subroutines, user exits, and utilities.
<i>Construct Spectrum Administrator’s Guide, variable name</i>	Italicized text indicates: <ul style="list-style-type: none"> • Book titles. • Placeholders for information you must supply.

Example	Description (continued)
<i>[variable]</i>	In syntax and code examples, values within square brackets indicate optional items.
{WHILE UNTIL}	In syntax examples, values within brace brackets indicate a choice between two or more items; each item is separated by a vertical bar ().

Related Documentation

The documentation sets for Construct Spectrum, Construct Spectrum SDK, and Natural Construct consist of the following manuals:

Construct Spectrum SDK

- *Construct Spectrum Programmer's Guide*
This guide is for developers creating Natural modules and ActiveX Business Objects to support applications that will run in the Natural mainframe environment and a Windows environment and/or an internet server.
- *Developing Web Applications*
This guide is for developers creating the web components of applications. It explains how to use the Construct Spectrum wizards in Visual Basic to generate HTML templates, page handlers, and object factory entries. It also contains detailed information about customizing, debugging, deploying, and securing web applications.
- *Construct Spectrum Messages*
This manual is for application developers, application administrators, and system administrators who wish to investigate messages returned by Construct Spectrum run-time and SDK components.
- *Developing Client/Server Applications*
This guide is for developers creating client components for applications that will run in the Natural mainframe environment (server) and a Windows environment (client).

Construct Spectrum

- *Construct Spectrum and SDK Client Installation*
This manual explains how to install and set up the Construct Spectrum run-time and SDK components on the client.
- *Construct Spectrum and SDK Mainframe Installation*
This manual explains how to install and set up the Construct Spectrum run-time and SDK components on the mainframe.
- *Construct Spectrum Administrator's Guide*
This guide is for administrators who wish use the Construct Spectrum Administration subsystem to set up and manage Construct Spectrum applications.

Natural Construct

- *Natural Construct Installation and Operations Manual for Mainframes*
This manual provides essential information for setting up the latest version of Natural Construct, which is needed to operate the Construct Spectrum programming environment.
- *Natural Construct Generation User's Manual*
This manual explains how to use the Natural Construct models to generate applications that will run in a mainframe environment.
- *Natural Construct Administration and Modeling User's Manual*
This manual explains how to use the Administration subsystem of Natural Construct and how to create new models.
- *Natural Construct Help Text User's Manual*
This manual explains how to create online help for applications that run on server platforms.

RUNTIME MESSAGES

This chapter describes the runtime error messages you may encounter when developing and running Construct Spectrum client/server applications.

The following topics are covered:

- **Introduction**, page 12
- **Resolving the Error**, page 13
- **Messages from the Spectrum Dispatch Client**, page 15
- **Messages from the Framework Classes**, page 23

For related information, see:

- *Entire Broker Messages and Codes*
- *Natural 2.2 Error Messages*
- **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*

Introduction

The Spectrum dispatch client and framework classes raise runtime error messages using the standard OLE Automation error mechanism. These errors can be trapped in Visual Basic with the On Error statement. You can then examine Visual Basic's Err object to obtain the error number, error source, and error description.

The following sections:

- provide information about resolving errors
- list the errors that originate in the Spectrum dispatch client
- list the errors generated by the framework classes

All error messages are listed in numerical order. Errors raised by the Spectrum dispatch client are in the 80040200 hex through 8004FFFF hex numeric range. For those readers familiar with OLE error codes, these numbers actually refer to an HRESULT. In an HRESULT, the high order bit (bit 31) indicates the severity, with 0 meaning success and 1 meaning failure. This bit is always 1 in an error situation. Bits 16 to 26 contain the facility code, which is always 4 (FACILITY_ITF), the code reserved for OLE interface errors. The low-order 16 bits indicate the actual error number, of which the range 00 hex to 1FF hex is reserved for use by OLE's Component Object Model.

All errors are in the range mentioned above.

Resolving the Error

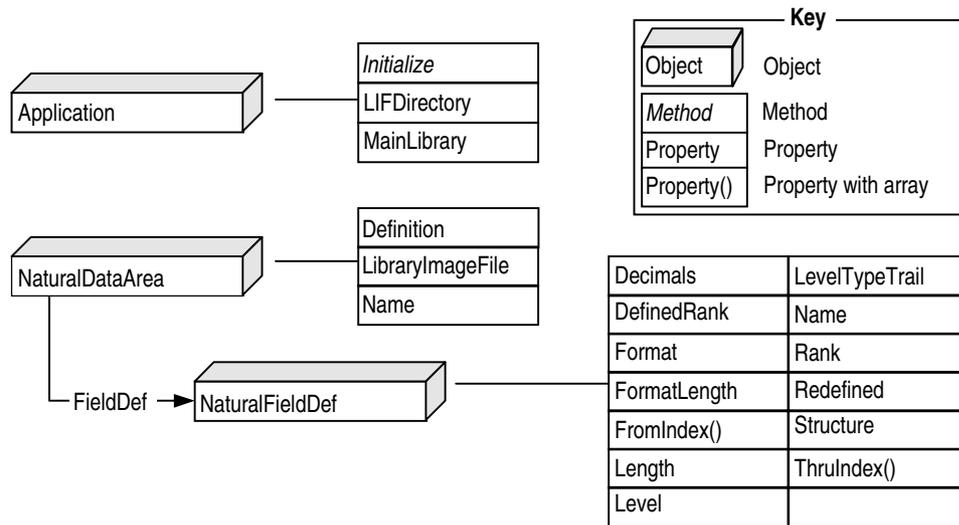
Resolving runtime errors is much easier if you know how to obtain information from the Spectrum dispatch client using the special properties and methods of its objects. For a detailed description of each object, property, and method mentioned in this section, see *Construct Spectrum Reference Manual*.

Checking Library Image File Definitions

Many common runtime errors occur due to differences between the library image file definitions and the code you write that uses those definitions.

For example, error 80040201 results when you refer to a field name that is not defined in the data area definition. Check the data area definition in the library image file and compare it to your code.

The following diagram shows the objects, properties, and methods you can use to obtain information about the definitions in a library image file:



The **Application** object searches for library image files in the LIF directory and uses the main library to obtain the steplib chain. These values are set with the *Initialize* method and can be queried with the *LIFDirectory* and *MainLibrary* properties.

When you are working with data area definitions, the **NaturalDataArea** and **NaturalFieldDef** objects have properties that return useful information about the data area definition, the library image file, and individual fields.

Messages from the Spectrum Dispatch Client

This section describes all Spectrum dispatch client runtime error messages in numerical order. Each error message number is shown in hex and decimal format, followed by a description of the error and suggestions on how to remedy the error.

80040200(-2147220992) The field syntax is invalid.

The field name does not have the correct syntax. One possible cause is that the field name contains a period character to separate the level 1 structure name from the field name, but the level 1 structure name is missing or there is a space before or after the period character.

80040201 (-2147220991) The field *field-name* is not defined in data area *data-area-name*.

The specified field does not exist in the data area definition.

Action Check the definition in the library image file to ensure that the field exists and the field name is spelled correctly.

80040202 (-2147220990) Ambiguous field name. The level 1 structure name is required to uniquely identify the field *field-name*.

The field name you are using occurs more than once in the data area definition in different level 1 structures.

Action Uniquely identify the field with the level 1 structure name.

80040203 (-2147220989) Incorrect number of indices. The field *field-name* has *count* dimensions.

Action Supply the correct number of index values when accessing the field. If the field is part of a structure array, provide an index value for each dimension of the field.

80040204 (-2147220988) Index value for dimension *dimension* is outside the defined range of *low-range* to *high-range* for field *field-name*.

Action Ensure that all index values are in the correct range for each dimension.

80040205 (-2147220987) The index specification is invalid.

Possible causes for this error are:

- More than three index values are provided.
- There is an opening parenthesis for the index values but no closing parenthesis.
- The upper index is less than the lower index in an index range, such as #A(4:1).

80040206 (-2147220986) The alphanumeric data being assigned to PackedData does not have the correct length. It must be *length* characters.

Action Check the NaturalDataArea.PackedDataLength property to determine the correct length of the alphanumeric data.

80040207 (-2147220985) Wrong number of parameter data areas. *Count* required.

When using the CallNat method of the Dispatcher object, you must pass one NaturalDataArea parameter for each level 1 field in the parameter data list.

Action Check the application service definition in the library image file for the number of level 1 fields required and change the code accordingly.

80040208 (-2147220984) Parameter *position* after the application service name is not of type NaturalDataArea.

All parameters to the CallNat method – except the first one – must be of type NaturalDataArea.

80040209 (-2147220983) The Variant data type (VB-data-type) of the value being assigned to the field is incompatible with the Natural format (Natural-format) of the field.

It is not possible to coerce the Variant data type into a value that can be stored in the Natural field.

Action Refer to *Construct Spectrum Reference Manual* to determine the compatibility between Variant data types and Natural formats.

8004020A (-2147220982) Numeric overflow.

The value assigned to the field is too large to be stored in the field.

8004020B (-2147220981) Invalid V substitution list.

The V substitution list in the Application.Allocate call is not valid for one of the following reasons:

- The substitution list must consist of one or more groups of parameters, where each group begins with a field name followed by the correct number of V substitution values for that field. There must be as many groups as there are fields with 1:V specifications.
- At least one of the substitution values is outside the range 1 to 32767.
- At least one of the substitution values is not a valid numeric data type.
- The same field name is specified more than once in the V substitution list.
- The data area definition does not contain any 1:V specifications, yet a V substitution list was specified.

8004020C (-2147220980) The data assigned to a structure must be an array of bytes.

The NaturalDataArea object reads and writes structures (and all fields within them) as byte arrays. This error occurs when you assign data to a structure, but the data is not a byte array.

8004020D (-2147220979) The size of the byte array (*array-size* bytes) does not match the size of the structure (*structure-size* bytes).

The NaturalDataArea object reads and writes structures (and all fields within them) as byte arrays. This error occurs when you assign a byte array to the structure, but the array does not have the correct size.

8004020E (-2147220978) The call parameters must be the same as the original parameters in retry mode.

The Retry property is True, but you are using a different set of parameters in the CallNat.

Action Either set Retry to False or reissue the same call with the same set of parameters.

8004020F (-2147220977) Cannot process multiple occurrences.

An index range was used where index ranges are not permitted. The only property or method that supports index ranges is the NaturalDataArea.Reset method.

80040210 (-2147220976) The definition for object *object-name* does not exist in the main library image file or its steplibs.

When the Application.Allocate method was in use, the data area definition could not be found. When the Dispatcher.CallNat method was in use, the application service definition could not be found.

Action Check the main library image file and its steplibs for the definition. For more information, see **Checking Library Image File Definitions**, page 13.

80040211 (-2147220975) Unable to open library image file *filename*. Error *error-number*, *error-message* occurred.

This error occurs whenever there is an unrecoverable error opening or reading the library image file. The substitution parameters provide additional information about the error.

80040212 (-2147220974) Abort or Commit called outside of a transaction.

The application called Abort or Commit when no transaction was active. The transaction may have been terminated by a timeout, by the subprogram proxy, or by your code earlier. When transactions are used, some errors implicitly end the transaction.

Action Ensure that the code explicitly checks all errors returned by the Dispatcher object so that the client application does not unnecessarily call Commit or Abort when no transaction is active.

80040213 (-2147220973) Unable to load a dynamic link library (*DLL-filename*) required by Spectrum. The following error occurred: *error-message*.

This error occurs when a DLL used by the Spectrum dispatch client cannot be loaded. This may be caused by one of the following reasons:

- The DLL is not on the PC.
- The DLL is on the PC but not in the Windows search path.
- The DLL has dependencies on other DLLs, which cannot be loaded.

Action Ensure that all required DLLs are installed correctly and in the Windows search path.

80040214 (-2147220970) The method *method-name* is not defined for application service *app-service-name*.

The application service definition in the library image file does not contain the method being used. When you use the `Dispatcher.CallNat` method and do not explicitly specify a method name, the Spectrum dispatch client uses a method name `DEFAULT`.

Action Ensure that the application service definition in the main library image file and its `steplibs` has the specified method name. For more information, see **Checking Library Image File Definitions**, page 13.

80040215 (-2147220969) Retry not possible after the last operation.

You may only set the `Retry` property to `True` if the request has been accepted by Entire Broker and an Entire Broker call returns an error with one of the following error classes: 36, 37, 74, or 215.

Action Check the `Dispatcher.RetryPossible` property to determine if you can set the `Retry` property and resume the request.

80040216 (-2147220968) This operation is not valid in retry mode.

When the `Retry` property is set to `True`, the following operations cannot be performed:

- Changing the `TraceOption` or `TraceCommand` properties.
- Calling `StartTransaction`, `Abort`, or `Commit`.

- 80040217 (-2147220967)** **The alphanumeric data *data* at offset *offset* cannot be mapped into a field with format *Natural-format*.**
- There is probably a parameter mismatch between the client application and the subprogram.
- Action** Ensure that the order of the parameters and parameter data areas in the client application matches the parameter data of the subprogram being called.
- 80040300 (-2147220736)** **Error in the data area definition, line *line-number*; the previous level is empty.**
- The data area definition has a structure or structure array that does not contain any elementary fields.
- 80040301 (-2147220735)** **Error in the data area definition, line *line-number*; invalid level change.**
- There is an inconsistency in the level numbers in the data area definition.
- 80040302 (-2147220734)** **Error in the data area definition, line *line-number*; the field being redefined does not exist.**
- A redefine in the data area definition refers to a field that does not exist.
- 80040303 (-2147220733)** **Error in the data area definition; the size of redefine *field-name* exceeds the size of the original field.**
- The size of the redefine is larger than the size of the original field.
- 80040304 (-2147220732)** **Error in the data area definition; field *field-name* has too many dimensions.**
- The data area contains a field that has more than three dimensions.

80040305 (-2147220731) Error in the data area definition, line *line-number*; FILLER used outside a redefine.

FILLER must only be used inside a redefine section.

80040306 (-2147220730) Error in the application service definition, line *line-number*; *message*.

There is an error in the application service definition. The message indicates what the error is.

80040307 (-2147220729) Error in the library image file, line *line-number*; unknown definition type *type*.

An unknown definition type was found in the library image file. The valid definition types are DataArea, AppService, and StepLibs.

80040308 (-2147220728) Error in the library image file, line *line-number*; invalid syntax.

This is caused by one of the following errors in the library image file:

- A definition header line does not end with a bracket.
- A DataArea or AppService definition header line does not contain a name.

80040309 (-2147220727) Error in the data area definition; field *field-name* has an invalid index range.

The specified field in the data area definition contains an invalid index range.

8004030A (-2147220726) Error in the data area definition; field *field-name* has too many significant digits. Maximum of 29 allowed.

The Spectrum dispatch client supports up to 29 significant digits.

Messages from the Framework Classes

This section describes runtime error messages raised by framework classes in the CSTVBFW DLL.

80040200 (-2147220992) Invalid modifier *modifier*.

The modifier keyword name was omitted. Each modifier must begin with a modifier keyword, followed by an equal sign (=), followed by the modifier value.

80040201 (-2147220991) The BDT *BDT-name* is not registered.

The BDT name has not been registered with the BDT controller. All BDTs used in the application must be registered with the BDT controller by calling the BDTController.RegisterBDT method.

80040202 (-2147220990) The Natural to BDT mapper is not registered.

When the BDT name is omitted in a call to ConvertToDisplay, ConvertFromDisplay, CreateSampleString, or ConvertInPlace, the Natural to BDT mapper function is used to choose a BDT. This error occurs because no Natural to BDT mapper function has been registered with the BDTController. RegisterNaturalBDTMapper method.

80040203 (-2147220989) You must provide either the BDT name or the Natural format/length.

A BDT name, Natural format string, or both must be provided when calling a BDT conversion routine.

80040204 (-2147220988) Invalid Natural format/length.

The Natural format string passed to a BDT conversion routine is not valid.

80040205 (-2147220987) The class *class-name* does not have a method called *method-name*.

When a call is made to the RegisterBDT or RegisterNaturalBDTMapper methods, the object and method name parameters must identify a valid method in the object's class.

80040206 (-2147220986) An unhandled runtime error occurred when calling the method *method-name* in object *class-name*: Error *error-number*, *error-message*.

An unhandled runtime error occurred in the BDT conversion routine mentioned in the error message. All BDT conversion routines must ensure that they use Visual Basic runtime error handling to trap any errors that might occur and return those errors in the error properties ErrorCode and ErrorMessage.

80040220 (-2147220960) Invalid registry hive name.

The registry hive name used in the assignment to the LanguageRegistryKey property must begin with HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, or HKEY_USERS.

80040221 (-2147220959) Invalid link identifier.

The link name in the resource file must include a resource ID and, optionally, a resource group ID and resource filename. If you are not using the optional components, you must use comma placeholders.

80040222 (-2147220958) Circular link.

The link name in the resource file links to other resources that eventually link back to the original resource, thereby forming a circular link.

80040223 (-2147220957) Invalid byte value *byte-value*.

Inline byte values in a binary resource must be specified with 1 or 2 hex digits for each byte.

80040224 (-2147220956) Empty byte resource.

An inline byte resource must have at least one byte specified.

80040225 (-2147220955) Invalid escape sequence *escape-sequence*.

The text resource includes a backslash to introduce an escape sequence; however, the characters that follow are not a valid escape sequence. Escape sequences are: nl, cr, lf, tb, or a sequence of three digits.

Action To include a backslash character in the text resource, use a sequence of two backslash characters.

80040240 (-2147220928) Attempt to add duplicate unique RowID *row-ID*.

Each row added to the BrowseDataRows cache must have a unique row ID. This error occurs if the cache already contains a row with the same row ID as the one being added.

Action Use a unique row ID for each row added to a generated local browse object using the INSERT-ROWS exit.

80040260 (-2147220896) Unable to determine control colors in *object-name*.

The control being added to an ObjectErrors collection does not have ForeColor or BackColor properties. These properties are set to the error validation color when a validation error occurs for the field the control is linked to.

80040261 (-2147220895) Invalid parameters passed to *object-name*.

An invalid set of parameters was passed to the Count or Remove method of an ObjectErrors collection. For more information about these methods, see *Construct Spectrum Reference Manual*.

80040280 (-2147220864) Unknown settings root type *root-type*.

The setting root type used in a call to Setting.Add, SettingList.Add, or SettingList.Read must be either SETTING_APPLICATION or SETTING_USER.

Action Use SETTING_APPLICATION to read and write application preferences that are independent of user. Use SETTING_USER to read and write user-specific settings.

80040281 (-2147220863) Invalid data type *data-type*.

The data type used in a call to Setting.Add or Setting.Update must be one of the following:

- SETTING_SZ, SETTING_BINARY
- SETTING_DWORD
- SETTING_DWORD_LITTLE_ENDIAN
- SETTING_DWORD_BIG_ENDIAN
- SETTING_MULTI_SZ

COMMUNICATION MESSAGES

This chapter describes the communication error messages you may encounter when developing and running Construct Spectrum client/server applications.

The following topics are covered:

- **Introduction**, page 28
- **Resolving the Error**, page 30
- **Communication Messages**, page 36

For related information, see:

- *Entire Broker Messages and Codes*
- *Natural 2.2 Error Messages*
- **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*

Introduction

Communication errors occur during a remote call from the client application to the subprogram. Many individual software components and data files are involved in each remote call. Some software components are running on the client, while others are running on the server. These include:

- Client application
- Spectrum dispatch client
- Library image files
- Entire Broker stub
- Entire Net-Work
- Entire Broker
- Spectrum dispatch service
- Spectrum security service
- Application service definitions
- Subprogram proxy
- Subprogram

Errors may originate in any one of these components, and each component may have its own predefined error codes. The Spectrum dispatch client returns the original, unchanged error code so that you can look it up in the appropriate documentation for that component.

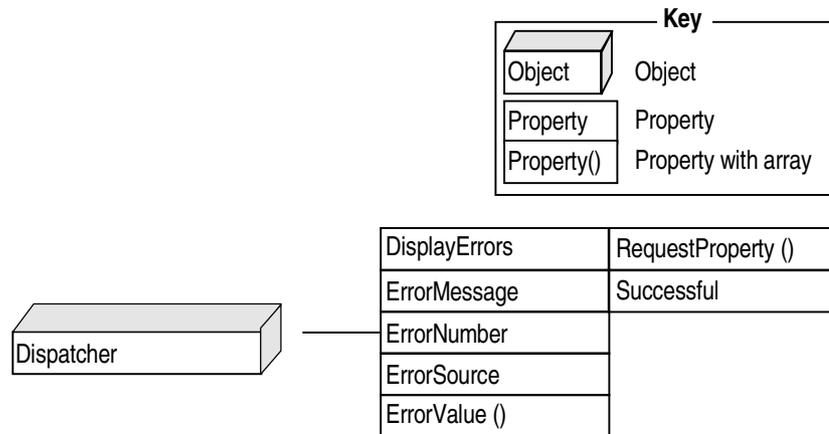
Since the client application is always the initiator of every remote call, it is also necessary to transfer back to the client application any error that does occur so that the client application can take corrective action or display the error message to the user. In all but the most severe error situations, all error messages are returned to the Spectrum dispatch client.

Severe errors that prevent the error message from being returned to the Spectrum dispatch client include the following:

- There was an interruption in the Entire Net-Work communication between the server and the client.
- Entire Broker timed out the conversation because the subprogram executed for a period longer than the Entire Broker conversation timeout value.
- The subprogram or one of the Spectrum services caused Natural to abend.

Resolving the Error

To obtain error or diagnostic information on communication errors, use the properties for the Spectrum dispatch object. The following diagram shows the Dispatcher object and its properties:



Communication errors are returned in the error properties of the Dispatcher object: ErrorMessage, ErrorNumber, ErrorSource, ErrorValue, and Successful. You can use other properties to indicate how error messages are displayed or return information about the last request.

Dispatcher Object Error Properties

The following sections describe each of the properties for the Dispatcher object.

DisplayErrors Property

This property indicates how communication error messages are displayed. For example, if DisplayErrors is True, the Spectrum dispatch client displays the error message in a message box.

ErrorMessage Property

This property returns the error message with substitution parameters in place.

ErrorNumber Property

This property (string) returns the error number. Its format and value depend on the error source. For example, if a runtime error occurs in a Natural service, Error-Number is the four-digit Natural error code. The error number format for each error source is:

Error Source	Error Number	Example
ETB	The four-digit error class followed by the four-digit error number.	02150148
NAT	The four-digit Natural error code.	0082
SPE	A four-digit error code.	5029

ErrorSource Property

This property indicates the component in which the error originated. Possible error sources are:

Error Source	Description
ETB	The error originated in the Entire Broker stub, Entire Broker, or Entire Net-Work.
NAT	The error is a Natural runtime error originating on the server.
SPE	The error originated in the Spectrum dispatch service, one of its subsystems, or the Spectrum dispatch client.

ErrorValue Property

This property (array) contains substitution values for the error message.

RequestProperty Property

This property returns information about the last request. It takes a property name and returns the value of that property as follows:

Property Name	Data Type	Description
Packet.CountOut	Integer	Number of outgoing packets sent to the Spectrum dispatch service.
Packet.DataOut(<i>x</i>)	Byte()	Data sent in outgoing packet <i>x</i> , where <i>x</i> ranges from 1 to Packet.CountOut value.
Packet.CountIn	Integer	Number of incoming packets received from the Spectrum dispatch service.
Packet.DataIn(<i>x</i>)	Byte()	Data received in incoming packet <i>x</i> , where <i>x</i> ranges from 1 to Packet.CountIn value.
Request.Domain	String	Dispatcher domain used.
Request.Object	String	Dispatcher object used.
Request.Version	String	Dispatcher version used.
Request.Method	String	Dispatcher method used.
Request.Subprogram	String	Subprogram name specified in the call.
Request.DataAreas	Integer	Number of data areas passed in the call.
Request.DataArea(<i>x</i>)	NaturalDataArea	NaturalDataArea object passed as parameter <i>x</i> .
Request.DataOut	Byte()	Complete request message.

Property Name	Data Type	Description (continued)
Request.DataIn	Byte()	Complete response message.
Request.BlocksOut	String	Alphanumeric data passed to the server for the level 1 blocks, preceded by the block header.
Request.BlocksIn	String	Alphanumeric data received from the server for the level 1 blocks, preceded by the block header.

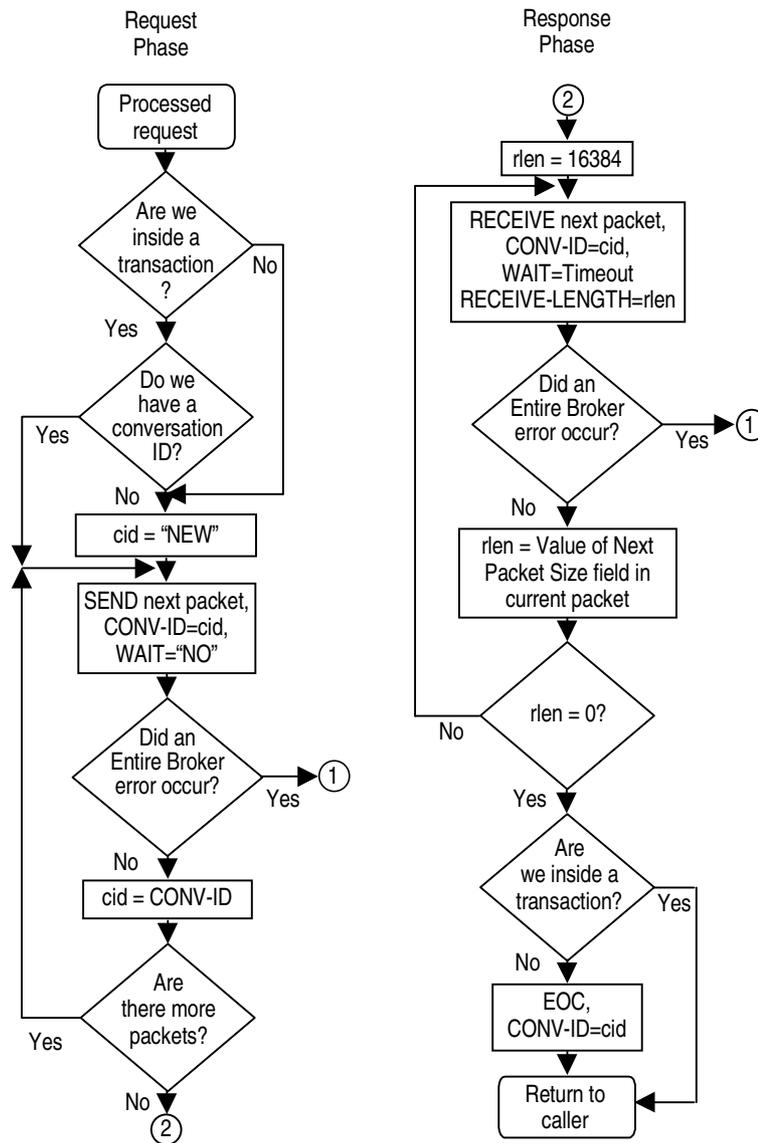
Successful Property

Depending on the outcome of the call, this property returns:

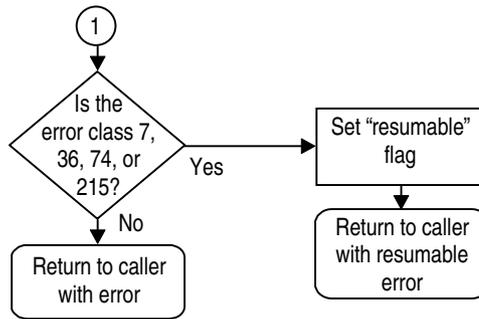
- True (call was successful)
- False (error occurred)

How Entire Broker Calls Are Used

When debugging the communications between the client and the Spectrum dispatch service, it is sometimes necessary to understand the Entire Broker calls that send the request data to the server and receive the response data from the server. The following two flowcharts illustrate this process:



Entire Broker Calls in the Request-Response Cycle



Entire Broker Error Handling

Communication Messages

This section describes all ETB and SPE Spectrum dispatch client communication errors in numerical order, grouped by error source. Each error includes an explanation and suggestions on how to remedy the error. For more information about Natural runtime errors, see **Natural Messages**, page 89.

Entire Broker Messages

Errors with this error source are always the result of an Entire Broker ACI call to the Entire Broker stub, where the ERROR-CLASS returned was not 0. If you need to know the ACI field values used in the Entire Broker call that resulted in the error, use RequestProperty to obtain them and use the property names beginning with “ETB.” These properties correspond to the ACI field with the same name.

This section describes Entire Broker errors that are common and how to resolve them in relation to Construct Spectrum. For a complete reference for all Entire Broker errors, refer to *Entire Broker Messages and Codes*.

ETB00070007

Service Not Registered.

The Spectrum dispatch service is not limited to a running attach server and all running services have either been stopped, timed out, or canceled.

Action Take the following steps:

- Use the Ping action on the Manage Services panel in the Spectrum Administration subsystem to check if the Spectrum service is running. If it is not running, start the Spectrum dispatch service.
- Ensure that the desired service is linked to an attach service so that it starts automatically upon request.
- Ensure that the associated attach service is executing by pinging it. Initiate the attach service if it is not running.
- If the attach server or desired server fails to start, check the communication log to see why it is failing.

ETB02150148**Entire Broker not active.**

The requested Entire Broker is not active or is no longer accessible through Entire Net-Work. A possible cause for this error is that some of the DLLs required by the Entire Broker stub or the Entire Net-Work client are not in the Windows path. By default, the installation program for Entire Net-Work V2.3.1 does not add the ADALNK and ETB\BIN directories to the Windows search path.

Action Take the following steps:

- Determine which Entire Broker is being requested by examining the details in the Network Error dialog or using the Spectrum Service Manager program.
- Use the Entire Net-Work Console to check whether the Entire Broker database ID is currently active.
- Ensure that the root directory for Entire Net-Work V2.3.1 is in the path, as well as the ADALNK and ETB\BIN subdirectories. The Readme file for Entire Net-Work mentions that you must do this before starting Entire Net-Work.

ETB00200216**API: Invalid BROKER-ID.**

The BROKER-ID field in the call does not contain a valid numeric node ID.

Action Take the following steps:

- Determine which broker ID is being used by examining the details in the Network Error dialog or using the Spectrum Service Manager program.
- Use the Spectrum Service Manager to correct the value of the BrokerID property.

Construct Spectrum Messages

SPE errors in the range 5500 to 5599 originate in the Spectrum dispatch client and usually occur because:

- there are Entire Broker or Entire Net-Work configuration problems
- or
- there is a mismatch between the data the Spectrum dispatch client is expecting to receive and the data that the Spectrum dispatch service sent

The RequestProperty property contains valuable information about the last request. Properties beginning with “Packet.” and “Request.” are useful.

SPE5500 **Unable to load a dynamic link library (*DLL-filename*) required by Spectrum. The following error occurred: *error-message*.**

This error occurs when a DLL used by the Spectrum dispatch client cannot be loaded. This may be caused by one of the following reasons:

- The DLL is not on the PC.
- The DLL is on the PC but not in the Windows search path.
- The DLL has dependencies on other DLLs, which cannot be loaded.

Action Check that all required DLLs are installed correctly and in the Windows search path.

- SPE5501** **The Spectrum dispatch service is using an unsupported packet protocol version: *packet-protocol-version*.**
- This error occurs if the data received from the Spectrum dispatch service contains an unsupported packet protocol version or has become corrupted. Data can become corrupted if the Spectrum Entire Broker services in the Entire Broker attribute file are defined with a translation routine using a line such as “TRANSLATE=SAGTCHA”.
- Action** Check that the Entire Broker attribute file does not define a translation routine for the Spectrum services. Otherwise, contact SAGA SOFTWARE support.
- SPE5502** **The packet header received from the Spectrum dispatch service is invalid: *packet-header*.**
- This error occurs if the data received from the Spectrum dispatch service contains an invalid packet header or has become corrupted.
- Action** Contact SAGA SOFTWARE support.
- SPE5503** **The Spectrum dispatch service is using an unsupported response protocol version: *response-protocol-version*.**
- This error occurs if the data received from the Spectrum dispatch service is using an unsupported response protocol version or has become corrupted. The most probable cause is that the Spectrum dispatch service has been updated and you are using an older version of the Spectrum dispatch client.

- SPE5504** **The Spectrum dispatch service did not return a valid message number in the response data: *response-data*.**
- This error occurs if the data received from the Spectrum dispatch service does not contain a valid Spectrum dispatch service message number or has become corrupted.
- Action** Contact SAGA SOFTWARE support.
- SPE5505** **The In/Out block header received from the Spectrum dispatch service is invalid: *block-header*.**
- This error occurs if the data received from the Spectrum dispatch service does not contain a valid in/out block header or has become corrupted.
- Action** Contact SAGA SOFTWARE support.
- SPE5506** **The TIMESTMP data is invalid.**
- This error occurs if the data received from the Spectrum dispatch service does not contain valid TIMESTMP data or has become corrupted.
- When using the TIMESTMP trace command, the Spectrum dispatch service appends timestamp data to the end of the response message. The Spectrum dispatch client expects to receive this data in a specific format, yet the data does not match that format.
- Action** Contact SAGA SOFTWARE support.

SPE5507

The Spectrum dispatch service did not send back the expected amount of data. *expected-size* bytes were expected but *actual-size* bytes were received.

This error occurs if the data received from the Spectrum dispatch service is either longer or shorter than expected. This error has the following causes:

- The data areas passed to the CallNat method do not match the parameter data of the receiving subprogram.
- The in/out block header does not correspond to the blocks that are actually sent by the subprogram proxy to the Spectrum dispatch client, because of an error in the subprogram proxy.
- Data has become corrupted.

Action Check the data area definitions for the parameter data areas passed to the CallNat method and match them against parameter data for the subprogram.

You can also use the Diagnostics program to determine the block names and sizes expected by the subprogram proxy. For information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.

SPE5508

The data received from the server does not match the parameter data area definition. The alphanumeric data *data* at offset *offset* cannot be mapped into a field with format *Natural-format*.

This error occurs if the data received from the Spectrum dispatch service is of the correct length, but the alphanumeric data received for a field is not in the correct format. This error has the following causes:

- The data areas passed to the CallNat method do not match the parameter data of the receiving subprogram.
- The in/out block header does not correspond to the blocks that are actually sent by the subprogram proxy to the Spectrum dispatch client because of an error in the subprogram proxy.
- Data has become corrupted.

Action

Check the data area definitions for the parameter data areas passed to the CallNat method and match them against parameter data for the subprogram.

You can also use the Diagnostics program to examine the data format expected by the subprogram proxy. For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.

SPE5509

The request or transaction has been canceled due to a time-out.

This error occurs in the following situations:

- When inside a dispatcher transaction, the Spectrum dispatch service timed out the transaction (or the Entire Broker CLIENT-NONACT timeout period was exceeded) because the client application did not send any further requests to the Spectrum dispatch service.
- When the Spectrum dispatch client displays the “Server is not responding” message and the user does not respond within the CLIENT-NONACT or CONV-NONACT timeout periods for Entire Broker.

- SPE5510** **The packet received from the Spectrum dispatch service is empty.**
- This error occurs if a packet received from the Spectrum dispatch service has a valid header, but the rest of the packet is empty.
- Action** Contact SAGA SOFTWARE support.
- SPE5511** **The Spectrum dispatch service did not send back a valid response: *response-data*.**
- This error occurs if the Spectrum dispatch service sends back an incomplete response.
- Action** Contact SAGA SOFTWARE support.
- SPE5512** **Unable to decrypt the response data.**
- This error occurs if the Spectrum dispatch client is unable to decrypt an encrypted response from the Spectrum dispatch service.
- Action** Contact SAGA SOFTWARE support.
- SPE5513** **The dispatch definition *definition-name* does not exist.**
- The dispatch definition name assigned to the Dispatcher.DispatchType property does not correspond to a valid dispatch definition in the Spectrum Service Manager.
- Action** Use a valid dispatch definition name.
- SPE5514** **User ID is missing.**
- The Application.UserID property in the Spectrum dispatch client has not been assigned a value. A user ID is required for all calls to the Spectrum dispatch service.
- Action** Assign a user ID to the Application.UserID property.

SPECTRUM SYSTEM MESSAGES

This chapter describes the Construct Spectrum system error messages you may encounter when developing and running Construct Spectrum client/server applications.

The following topics are covered:

- **Introduction**, page 46
- **Entire Broker Messages**, page 47
- **Construct Spectrum Messages**, page 54
- **Natural Messages**, page 89

For related information, see:

- *Entire Broker Messages and Codes*
- *Natural 2.2 Error Messages*
- **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*

Introduction

The server components of the Spectrum Administration subsystem generate error messages that require developer or administrator intervention. Correcting these errors usually involves updating the information in the Spectrum Administration subsystem or modifying the Entire Broker or Entire Net-Work configuration.

An error marked with “NS” denotes an error generated because Natural Security is active.

The sections in this chapter are organized according to the following message type:

- Entire Broker error messages
- Construct Spectrum error messages
- Natural error messages

All other error messages are generated by Natural Construct-generated code or the Natural runtime environment. For information about these error messages, refer to *Natural 2.2 Error Messages*.

Entire Broker Messages

This section describes common Entire Broker error messages and how to resolve them in relation to Construct Spectrum. For more information about Entire Broker error messages, refer to *Entire Broker Messages and Codes*.

Error Class 0007

ETB00070007**Service Not Registered.**

The Spectrum dispatch service is not linked to a running attach server and all running services have either been stopped, timed out, or canceled.

Action Take the following steps:

- Examine the details in the Network Error dialog (or use the Dispatch Service Manager program) to determine which service is being requested.
- Ping the dispatch service on the Manage Services panel (Spectrum Administration subsystem) to determine if it is running. If it is not running, initiate the dispatch service.
- Specify the attach service name to ensure that the desired service is linked to an attach service and starts automatically upon request. In addition, mark the Attach field. If attach servers are running, they must be refreshed to activate recent file changes.
- Ping the associated attach service to determine if it is executing. If it is not running, initiate the attach service.
- If the attach server or desired server fails to start, check the communication log to see why it is failing.

Error Class 0020

User error in the API.

ETB00200201 **API> Unknown Broker ID *broker ID* defined in service *service name*.**

The specified broker ID does not match the broker ID defined in the Entire Broker attribute file.

Action Take one or more of the following steps:

- Check the value in the Broker ID field on the first Maintain Services panel (Spectrum Administration subsystem) for each defined service. Ensure that it matches the value defined in the attribute file.
- If the broker ID is assigned in code, ensure that the specified value also matches the attribute file.

ETB00200212 **API> (*) in service *server class / server name / service* is invalid for function.**

The definition of the Entire Broker services contains invalid data. For a Send function, server class, server name, and service values cannot contain an asterisk.

Action Take one or more of the following steps:

- Check the values in the Server Class, Server Name, and Service fields on the first Maintain Services panel (Spectrum Administration subsystem) for each defined service. Ensure that the fields do not contain asterisks.
- If these values are assigned in code, ensure that the specified values do not contain asterisks.

ETB00200216 **API> Invalid Broker ID *broker ID* defined in service *service name*.**

The broker ID must have a three-digit node ID (Adabas DBID) at the end of an arbitrary text. The node ID is defined in the Entire Broker attribute file. For example: BKR045, B201, or ETB111.

- Action** Take one or more of the following steps:
- Check the value in the Broker ID field on the first Maintain Services panel (Spectrum Administration subsystem) for each defined service. Ensure that it matches the value specified in the attribute file.
 - If the value is assigned in code, ensure that the specified value matches the attribute file.

Error Class 0021

Configuration error in the Entire Broker attribute file.

ETB00210015 **ATTR> Maximum possible number of clients reached.**

The NUM-CLIENT value defined in the Entire Broker attribute file has been exceeded. By default, Construct Spectrum attempts the request as many as five times when this error is received. The error message is only returned if all five attempts fail.

- Action** Increase the value in the NUM-CLIENT field (up to a maximum of 6,000 clients).

ETB00210018**ATTR> Maximum possible number of servers reached.**

The NUM-SERVER value defined in the Entire Broker attribute file has been exceeded. By default, Construct Spectrum attempts the request as many as five times when this error is received. The error is only returned if all five attempts fail.

Action Increase the value in the NUM-SERVER field (up to a maximum of 2,000 servers).

ETB00210043**ATTR> Service *server class / server name / service* definition not found.**

The initiated Spectrum service is defined with a server class, server name, and service that are not defined in the Entire Broker attribute file.

Action Do one of the following:

- Change the service values on the Maintain Services panel (Spectrum Administration subsystem) to match the values defined in the attribute file.

Or

- Add the server class, server name, and service to the attribute file. Ensure that the command service (suffixed by CMD) is also defined in the file.

ETB00210057**ATTR> No more available short message buffers.**

The NUM-SHORT-BUFFER value defined in the Entire Broker attribute file has been exceeded. By default, Construct Spectrum attempts the request as many as five times when this error is received. The error is only returned if all five attempts fail.

Action Increase the value in the NUM-SHORT-BUFFER field (up to a maximum of 999,999).

- ETB00210061** **ATTR> No more available long message buffers.**
- The NUM-LONG-BUFFER value defined in the Entire Broker attribute file has been exceeded. By default, Construct Spectrum attempts the request as many as five times when this error is received. The error is only returned if all five attempts fail.
- Action** Increase the value in the NUM-LONG-BUFFER field (up to a maximum of 999,999).
- ETB00210096** **ATTR> Maximum possible number of services reached.**
- The NUM-SERVICE value defined in the Entire Broker attribute file has been exceeded. By default, Construct Spectrum attempts the request as many as five times when this error is received. The error is only returned if all five attempts fail.
- Action** Increase the value in the NUM-SERVICE field (up to a maximum of 1,000).
- ETB00210230** **ATTR> Maximum possible number of conversations reached.**
- The NUM-CONVERSATION value defined in the Entire Broker attribute file has been exceeded. By default, Construct Spectrum attempts the request as many as five times when this error is received. The error is only returned if all five attempts fail.
- Action** Increase the value in the NUM-CONVERSATION field (up to a maximum of 10,000).

Error Class 0074

Timeout message errors.

ETB00740074

Wait timeout of *number of seconds* has been exceeded.

The request failed because the WAIT time specified for the receive function has elapsed. This may be caused by lack of services, abended services, or system load.

Action Do one of the following:

- 1 Retry the action.
- 2 If the message recurs, add new services using the Initiate function (Manage Services panel in the Spectrum Administration subsystem).

Or

- 1 Wait until system load is lower.

Error Class 0079

Error in the Entire Broker stub.

ETB00790214

STUB> Work space allocation failed - requires 33K.

This error is returned if this amount of memory is not available. Entire Broker requires a minimum of 33K of memory.

Action Reduce the ESIZE, FSIZE, DATSIZE, CSIZE, and/or ASIZE requirements for Natural.

Error Class 0215

Connection errors related to Entire Net-Work.

ETB02150148

Nucleus for *broker ID* is not running.

The specified broker ID cannot be reached. This may be because the Entire Broker is not running or Entire Net-Work is not aware of the required Entire Broker nucleus.

Action

Take the following steps:

- Ensure that the broker ID is running.
- Ensure that the system has access to the required ID, either directly or by means of Entire Net-Work.

Construct Spectrum Messages

This section describes the error messages used by Construct Spectrum, includes informational messages.

SPE5000	Security cache & cleanup queue are now synchronized.
Information only	Reflects the synchronization of updates made to users, domains, steplib, application service definitions, and security definitions in the security cache.
SPE5001	System has stopped 1 copy of Service <i>service name</i>.
Information only	Construct Spectrum located a copy of the service and successfully issued a shutdown request.
SPE5002	Server not started; Service <i>service name</i> not found.
	The specified Spectrum service cannot be found on file. Usually, this error is due to an incorrect LFILE setting for LFILE 136.
	When starting up, the Spectrum service executes an INPUT parameter indicating which service record to use. The service name is one of the user parameters on the record.
Action	Ensure that the LFILE used by the start task matches the one used by the attach server.

- SPE5003** **No *service name* services are available.**
- An attempt to access the Spectrum service failed. This may occur if:
- The specified service is not currently running.
 - The Entire Broker settings on the service record were updated since the service was started.
- Action** Take the following steps:
- 1 Start a copy of the service (on the Manage Services panel in the Spectrum Administration subsystem).
 - 2 If the service does not start, ensure that the specified Entire Broker settings values were not updated. If they were, return them to the values used to start the Spectrum service.
 - 3 Communicate with the running Spectrum service.
- SPE5005** **PING of Service *service name* was successful.**
- Information only** The request to locate at least one idle copy of the service was successful.
- SPE5006** **Maximum object size of *maximum size* characters exceeded.**
- The largest data stream that can be handled by the Spectrum dispatch service is 32300 bytes.
- Action** Change the size of the client component so the object does not exceed this number.
- SPE5008** **Last position in value must be " ", "H", "M" or "S".**
- The last character in the service timeout value must be blank, H (hours), M (minutes), or S (seconds).
- Action** Change the service timeout value so that it ends with a valid value.

- SPE5009** **Invalid format specified, must be 99999 or 9999X.**
The format of the value is invalid.
- Action** Change the value to be either all numeric or a 4-digit number followed by one of the valid suffixes defined in SPE5008.
- SPE5010** **CDAPROXY Object value of *object* invalid for system function request.**
The value specified in the Object field of the CDAPROXY data area is not supported as a SYSTEM function. This will only occur if the DOMAIN value in CDAPROXY has a value of SYSTEM.
- Action** Correct the code to populate the Object field with a valid value.
- SPE5011** **JCL member *Natural member* does not exist in library *Natural library*.**
The Natural member containing the required JCL cards was not found.
- Action** If the error occurred while a service was starting, ensure that the value associated with JCL=Keyword in the service start parameters exists as a Natural source member in the SYSSPEC library.
- SPE5012** **JCL in member *Natural member* contains lines GT *maximum JCL line length* characters.**
The source member used to store the JCL contains lines that exceed the specified maximum length (80 characters).
- Action** Update the contents of the Natural source member to eliminate any lines of JCL that exceed 80 characters.

- SPE5013** **Response code *error code* received from called routine.**
- The routine called to submit the JCL experienced an error.
- Action** Take the following steps:
- Check the parameters for the NATRJE call.
 - If these appear to be correct, contact your Natural administrator to ensure that the environment is properly defined to support NATRJE functionality.
- SPE5014** **Library must be specified for library listing function.**
- A library listing was requested by the Visual Basic Add-In, but the library name was not specified.
- Action** Contact SAGA SOFTWARE support.
- SPE5015** **The number of lines requested may not exceed *maximum lines*.**
- A request made by the Visual Basic Add-In has specified more than the maximum number of lines.
- Action** Contact SAGA SOFTWARE support.

SPE5016 **Error *Natural* error calling routine *Natural* module specified in debug member *debug* source.**

The debugging mechanism supported by the Invoke Proxy function triggered a Natural runtime error while accessing the module specified in the debug member.

Action Take the following steps:

- 1 Ensure that the values specified for Debug Library, Debug Member, DBID, and FNR are correct.
- 2 Ensure that the subprogram proxy specified in the Debug member exists in the specified location.
- 3 Ensure that Steplib Info in the Debug member is correct.
- 4 Ensure that the input parameters to the target module have not been altered. They should appear as follows:

```
PARAMETER
01 FORMAT-BYTE (I1)
PARAMETER USING CDAPROXY
PARAMETER USING CDPDA-M2
PARAMETER
01 #PDA (A1/1:32000)
```

SPE5017 **Test using specified debug data completed without errors.**

Information only The subprogram proxy was invoked based on the information found in the specified debug member. Testing was completed with no errors from the proxy or the Natural runtime environment.

SPE5018**No security servers are running; contact system administrator.**

All security lookups required by a Spectrum dispatch service are routed to a remote security service. If the dispatch service attempts to access its designated security service and fails because the security service is not running, this message is generated.

Action

Take the following steps:

- 1 Access the first Maintain Services panel (Spectrum Administration subsystem) for the Spectrum service definition. Note the value in the Security field.
- 2 Access the Manage Services panel and start a copy of that Spectrum security service.
- 3 Ensure that the security service is linked to an attach server.

SPE5019**Error *Natural error* occurred trying to access DBID Database ID and FNR System file number.**

The debugging mechanism supported by the Invoke Proxy function triggered a Natural runtime error while attempting to use the specified database ID (DBID) and system file number (FNR) values.

Action

Ensure that valid values are specified for DBID and FNR.

SPE5021 **No security privileges are available for user *user ID*.**

None of the groups to which the specified user belongs has privileges defined for any method of any object belonging to the specified domain.

Action Take one of the following steps:

- Find a group associated with the requested domain and add this group to the user definition.

Or

- Add a new group-domain combination to the security file for at least one of the groups to which the user belongs; assign the necessary privileges. For more information, refer to **Setting Construct Spectrum Security Options** in *Construct Spectrum Administrator's Guide*.

SPE5022 **User *user id* access privileges are revoked for *domain + object + method*.**

Privileges for the specified user are revoked for the specified request.

Action Check the validity of the request and take one of the following steps:

- If the request is incorrect, change the appropriate specification.

Or

- If the request is correct and access is required, ask the Spectrum Application administrator to assign the proper privileges.

SPE5023**User *user ID* access privileges are disabled for *domain + object + method*.**

Privileges for the request made by the specified user are disabled. This may occur when a system is undergoing maintenance.

Action Check the validity of the request and take one of the following steps:

- If the request is incorrect, change the appropriate specification.

Or

- If the request is correct and access is required, contact the Spectrum Application administrator to determine when the service will become available.

SPE5025**Domain *domainID* is not defined.**

The specified domain is not in the Construct Spectrum files.

Action Check the validity of the domain and take one of the following steps:

- If the domain is incorrect, change the appropriate specification.

Or

- If the domain is correct, ask the Spectrum Application administrator to add the domain.

SPE5026**Service Definition *domain* + *object* + *version* is not defined.**

The specified application service definition is not in the Construct Spectrum files.

Action

Check the validity of the application service definition (domain, object, and version) and take one of the following steps:

- If the application service definition is incorrect, change the appropriate specification.

Or

- If the definition is correct, ask the Spectrum Application administrator to add the application service definition.

SPE5027**Method *method name* is not defined in Service Definition *domain* + *object* + *version*.**

The specified method cannot be found for the specified application service definition.

Action

Check the validity of the method and application service definition (domain, object, and version) and take one of the following steps:

- If the method or application service definition is incorrect, change the appropriate specification.

Or

- If these are correct, ask the Spectrum Application administrator to add the method to the appropriate application service definition.

- SPE5028** **User ID *user ID* is not defined.**
- The specified user ID is not in the Construct Spectrum files.
- Action** Check the validity of the user ID and take one of the following steps:
- If the user ID is incorrect, change the appropriate specification.
- Or
- If the user ID is correct, ask the Spectrum System administrator to add the user ID.
- SPE5029** **Password for user *user ID* not valid.**
- The specified password for the specified user does not match the password in the Spectrum Administration subsystem.
- Action** Change the password and repeat the request. If you do not remember your password, have the Spectrum System administrator override your current password with a new password. Once this is completed, use the Change Password program installed with Construct Spectrum to update the password.
- SPE5030** **User *user ID* not defined to any groups; access denied.**
- The specified user does not belong to any group, so a security check cannot be performed.
- Action** Add at least one group to the User definition in the Spectrum Administration subsystem.
- SPE 5031** **SAF error:1:occurred.**
- SAF returned the specified error.
- Action** Refer to the SAF documentation.

- SPE5033** **Invalid User ID/Password supplied by security server.**
- The Spectrum security service client received an invalid internal User ID from its target server. This tells the Spectrum security service client that the server that handled its request was an invalid target server.
- Action** Ensure that the server module used for security (SRVRSECS) is the routine that was provided with the Spectrum Administration subsystem. If this error occurs, the target security service has been compromised. Delete the SRVRSECS module from the SYSSPEC system and re-install this module from the Construct Spectrum installation tape.
- SPE5034** **Default language must be in the range 1:60.**
- When defining services, a default language number must be specified. This value indicates which default language to use when communicating messages back to clients. Clients can override this value with each request, but the service always returns to this default once any request is completed.
- Action** Ensure that the value falls in the correct range. For a list of valid values and their associated languages, see **Language System Variable** in *Natural 2.2.x Reference Manual*.
- SPE5035** **Old password value for *user ID* not valid; password change denied.**
- The request to change the specified user password failed because the specified current password does not match the one found for the specified user.
- Action** Change the old password value and repeat the request. If you do not know the password value, have the Spectrum System administrator override your current password with a new password.

SPE5036**TARGET SUBPROGRAM INFORMATION section not found in *Natural source member*.**

The specified debug member in the Invoke Proxy function does not contain the block of information required to determine the name of the target subprogram proxy and its location.

Action

Take the following steps:

- 1 Ensure that the parameters specified on the Invoke Proxy panel are correct.
- 2 Ensure that the contents of the debug member have not been altered by validating the information presented by the List Directory function in Natural.
- 3 If the debug member was altered, regenerate it using the debugging facilities on the client. For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.

SPE5037**CDAPROXY CONTENTS - IN section not found in *Natural source member*.**

The specified debug member in the Invoke Proxy function does not contain the block of information required to define the specified service (domain, object, method, etc.).

Action

Take the following steps:

- 1 Ensure that the parameters specified on the Invoke Proxy panel are correct.
- 2 Ensure that the contents of the debug member have not been altered by validating the information presented by the List Directory function in Natural.
- 3 If the debug member was altered, regenerate it using the debugging facilities on the client. For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.

SPE5038 **DATA STREAM CONTENTS - IN section not found in *Natural source member*.**

The specified debug member in the Invoke Proxy function does not contain the block of information required to create the data values that must be sent to the target subprogram proxy.

- Action** Take the following steps:
- 1 Ensure that the parameters specified on the Invoke Proxy panel are correct.
 - 2 Ensure that the contents of the debug member have not been altered by validating the information presented by the List Directory function in Natural.
 - 3 If the debug member was altered, regenerate it using the debugging facilities on the client. For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.

SPE5039 **Messages for *service queue* displayed successfully.**

Information only The request to view messages for the Service Log completed without error.

SPE5043 **Control Record modified successfully.**

Information only The update to the Construct Spectrum control record completed successfully. Changes to Natural Security activation, the Entire Net-Work timeout, and the security exit were committed to the database.

SPE5044 **Requested *number of lines requested* line numbers, only *number of lines found* are available.**

A request made by the Visual Basic Add-In specified more than the available number of lines for a source code lookup.

- Action** Contact SAGA SOFTWARE support.

- SPE5045 (NS)** **User ID *user ID* not defined to Natural Security.**
The specified user is not defined to Natural Security.
- Action** Check the validity of the specified user ID and take one of the following steps:
- If the user ID is incorrect, change the appropriate specification.
- Or
- If the user ID is correct, ask the Natural Security administrator or Spectrum System administrator to add the user ID.
- SPE5046 (NS)** **User ID *user ID* is defined as a Group in Natural Security; disallowed.**
The specified user is defined as a Group type user ID. A Group type user cannot access system data.
- Action** Check the validity of the specified user ID and take one of the following steps:
- If the user ID is incorrect, change the appropriate specification.
- Or
- Use an A or P type user ID or change the user type in NASEC.
- SPE5047** **Module *Natural module* cannot handle request for *number of lines* lines.**
The specified Visual Basic Add-In requested more lines of source code than the routine can handle.
- Action** Contact SAGA SOFTWARE support.
- SPE5048** **Module *Natural module* can't handle requested lines of length *line length*.**
The specified Visual Basic Add-In requested source code lines of a length longer than the routine can handle.

- Action** Contact SAGA SOFTWARE support.
- SPE5049** **Member name must be specified.**
The Visual Basic Add-In requested the download of a Natural module, but did not specify the module name
- Action** Contact SAGA SOFTWARE support.
- SPE5050** **Library name must be specified if not using steplib.**
The Visual Basic Add-In requested the download of a Natural module, but did not specify the name of the library in which the module is stored.
- Action** Contact SAGA SOFTWARE support.
- SPE5052** **CMTASK error *error number* - no storage for parameter areas.**
Can be returned by the attach service if the CMTASK routine does not have access to the required memory.
- Action** Contact your System administrator.

- SPE5054** **Default security message - no MSG-NR supplied in *security exit*.**
- The Spectrum dispatch service expects one of two things from the security exit: grant access or a message number and substitution values. One of the non-grant situations in the security exit did not properly assign this message data.
- Action** Check the validity of the security exit specified on the Construct Spectrum Control record and take one of the following steps:
- If the security exit is incorrect, change the appropriate specification.
- Or
- If the security exit is correct, check the code in the exit. For all situations in which access is not granted, assign a message number and message data.
- SPE5055 (NS)** **User *user ID* not linked to library *Natural library* found on steplib *steplib record* (Natural Security).**
- One of the steplibs required to satisfy the specified user request is not linked to the user in Natural Security. The library may be any library in the steplib chain.
- Action** Take one of the following steps:
- Copy or move the target routine to a library that is in the steplib chain and linked to the user in Natural Security.
- Or
- Ask the Natural Security administrator or Spectrum System administrator to link the user to the necessary libraries.

- SPE5056** **Packet protocol using version *version number* and release *release number* was invalid.**
- The specified protocol (used to transmit the packet from the client to the Spectrum dispatch service) is not known by the Spectrum dispatch service.
- Action** Ensure that the packet protocol on the client is set correctly and sending the request to the correct Spectrum dispatch service.
- SPE5057** **Request protocol using version *version number* and release *release number* was invalid.**
- The specified protocol (associated with the final assembled request message on the server) is not known by the Spectrum dispatch service.
- Action** Ensure that the request protocol on the client is set correctly and sending the request to the correct Spectrum dispatch service.
- SPE5058** **Member *Natural source member* not found in library *Natural library*.**
- The Visual Basic Add-In requested the download of a Natural module from a specified library, but the specified module is not in the library.
- Action** Contact SAGA SOFTWARE support.
- SPE5059** **Number of lines requested must be greater than zero.**
- The Visual Basic Add-In requested the download of a negative number of source code lines.
- Action** Contact SAGA SOFTWARE support.

- SPE5060** **Call to decompression routine *routine name* failed with error *error number*.**
- The specified decompression routine did not complete successfully.
- Action** Take the following steps:
- 1 Ensure that the decompression routine (EXPND) is available to the specified Natural nucleus.
 - 2 Ensure that the specified routine is the one supplied with the current installation of Construct Spectrum.
- SPE5061** **Call to compression routine *routine name* failed with error *error number*.**
- The specified compression routine did not complete successfully.
- Action** Take the following steps:
- 1 Ensure that the compression routine (CMPRS) is available to the specified Natural nucleus.
 - 2 Ensure that the specified routine is the one supplied with the current installation of Construct Spectrum.
- SPE5062** **Call to encryption routine *routine name* failed with error *error number*.**
- The specified encryption routine did not complete successfully.
- Action** Take the following steps:
- 1 Ensure that the encryption routine (CSTCSN) is available to the specified Natural nucleus.
 - 2 Ensure that the specified routine is the one supplied with the current installation of Construct Spectrum.

SPE5063 **Call to decryption routine *routine name* failed with error *error number*.**

The specified decryption routine did not complete successfully.

- Action** Take the following steps:
- 1 Ensure that the decryption routine (CSTCSD) is available to the specified Natural nucleus.
 - 2 Ensure that the specified routine is the one supplied with the current installation of Construct Spectrum.

SPE5064 **Invalid function of *function* and method of *method* supplied to *Natural module*.**

The Visual Basic Add-In request resulted in an invalid request by the host-based proxies that support the Visual Basic Add-In.

- Action** Contact SAGA SOFTWARE support.

SPE5065 **Request can't be handled while cache cleanup in progress.**

Specified request was made while cache cleanup was in progress. All changes to users, domains, steplib, application service definitions, methods, and security privileges update the security cache.

- Action** Take one or more of the following steps:
- Wait for 15 seconds and attempt the request again. You can repeat the request several times.
 - If the message persists, ask the Spectrum System administrator to submit the Synchronize Security Cache & Cleanup Queue function. This job runs until the security cache is fully synchronized. Once this request completes, all requests should be active.
 - If the message persists, contact SAGA SOFTWARE.

- SPE5066** **Client's security mode must be set to:1:**
- The security mode setting for the dispatch service on the client does not match the security mode setting on the server.
- Action** Change the security mode setting on the client to match the security mode setting on the server.
- SPE5067** **One copy of *service name* started as attached task under *TP monitor*.**
- Construct Spectrum initiated the requested Spectrum service using the specified service start routine. In this case, the Spectrum service was started as an attached task under the current TP monitor.
- Action** Check the Communication Logs for messages generated by the initiated task. If no messages are available, the task started successfully. Error messages may take some time to appear, depending on how long it takes for the service to attempt a startup. Otherwise, you can find the error that caused the service to abort in this query.
- SPE5068** **Test for service *service name* completed successfully.**
- The service start routine showed only the results of the User parameters parsing function. No Spectrum service was started.
- Action** If errors were reported by the test function, correct the errors by modifying the User parameters field on the first Maintain Services panel in the Spectrum Administration subsystem.

- SPE5069** **Error *Natural runtime error occurred in service start routine Natural module at line line number.***
- The Construct Spectrum system attempted to call the service start routine specified on the Service record, but the service start routine failed with a runtime error.
- Action** Check the line of code specified and correct the error based on the runtime error specified.
- SPE5070** **Attach manager *service name* required by service *service name* not defined.**
- The value associated with the ATTACH= parameter in the User parameters of the Service definition cannot be found.
- Action** Ensure that the value associated with the ATTACH= parameter specifies the correct Spectrum service.
- SPE5073** **User parameter keyword *keyword* has no value for service *service name*.**
- Each keyword specified in User parameters fields on a service definition must have an associated value.
- Action** Ensure that each keyword specified in User parameters fields has an associated value.
- SPE5074** **Service start handler must be specified prior to testing.**
- The service start handler was not specified. The Maintain Services panels in the Spectrum Administration subsystem have a special PF-key to test parse the contents of the User parameters string. This testing function is only available when a service start routine is specified.
- Action** Specify the name of the service start routine.

SPE5076***Service name may not have embedded blanks.***

The name of the Spectrum service contains spaces, which is not allowed.

Action Change the service name to remove the spaces. You can replace the spaces with other characters, such as dashes (-) or underscore (_) characters.

SPE5077***Group *group ID* not found in Natural Security.***

This message indicates that the group has not been defined to Natural Security.

Action Check the validity of the specified group ID and take one of the following steps:

- If the group ID is incorrect, change the appropriate specification.

Or

- If the group ID is correct, ask the Natural Security administrator or Spectrum System administrator to add the group.

- SPE5078** **Security service exit is required if security not disabled.**
- If a Spectrum dispatch service is designated as secure, then a security service exit is required.
- Action** If you want the Spectrum dispatch service to be unsecured, mark the Disable Security flag on the second Maintain Services panel in the Spectrum Administration subsystem.
- If you want the Spectrum dispatch service to be secured, specify the security service exit. To populate this field:
- Display an existing secured Spectrum dispatch service, modify the contents of the service definition, and save it as a new service.
- Or
- Clear the current Spectrum dispatch service and request defaults (PF4). This request automatically fills in the Security field. Finish specifying the values and save the new Spectrum dispatch service.
- SPE5080** **Only select one option: online OR batch.**
- More than one option was selected. When deleting the security cache contents, Construct Spectrum counts the number of entries in the cache. You must specify whether to delete the cache in batch or online. This message is displayed if both options are selected.
- Action** Select only one of the deletion options.
- SPE5081** **Select at least one option: online OR batch.**
- No option was selected. When deleting the security cache contents, Construct Spectrum counts the number of entries in the cache. You must specify whether to delete the cache in batch or online. This message is displayed if neither option is selected.
- Action** Select at least one of the deletion options.

- SPE5082** **Security cache cleanup not initiated; exited with *function key*.**
- Information only** The request to clear the security cache was not performed; the request was terminated by a PF-key.
- SPE5083** **Batch job to reset Security Cache submitted successfully.**
- Information only** The job to clear the security cache in batch was successfully submitted and is running with a JOB name of CACHERST.
- SPE5084** **Security cache reset successfully; *number of record records deleted*.**
- Information only** If the cache clearing task was running online, this message is displayed after the job is completed. If the job was executed in batch, this message is written to the output queue of the job.
- SPE5085** **Security service is required if security not disabled.**
- If a Spectrum dispatch service is designated as secure, a security service is required.
- Action** If you want the Spectrum dispatch service to be unsecured, mark the Disable Security field on the second Maintain Services panel in the Spectrum Administration subsystem.
- If you want the Spectrum dispatch services to be secured, specify the name of the security service in the Security field on the first Maintain Services panel.

- SPE5091** **Service *service name* submitted as batch job *job name*.**
- Construct Spectrum initiated the requested Spectrum service using the specified service start routine. In this case, the Spectrum service was started as a batch job using NATRJE.
- Action** Check the Communication Logs for messages generated by the initiated task. If no messages appear, the task started successfully. Error messages may take time to appear, depending on how long it takes for the Spectrum service to attempt a startup. Otherwise, you can find the error that caused the Spectrum service to abort in this query.
- SPE5093** **Remote Services cannot be specified for *service name*.**
- You cannot specify remote services for type S (security) service definitions.
- Action** Take one of the following steps:
- Change the Spectrum service to a type other than S.
- Or
- Remove the value from the Security field on the first Maintain Services panel in the Spectrum Administration subsystem.
- SPE5094** **Attach Manager *service name* needed to start *service name* is unavailable.**
- There may be attach services running, but not the one expected by the specified Spectrum service. The service may require an attach service to start. If the attach service is not running, you cannot issue the start request.

Action Check the attach service value. If it is correct, start the required service. Once you can Ping the attach service, start the Spectrum service again.

If the value is incorrect, specify the required attach service and start the Spectrum service.

SPE5095 **Request sent to *attach* service to start one copy of service *service name*.**

Construct Spectrum initiated the requested Spectrum service using the specified service start routine. In this case, the request was routed through Entire Broker to an attach service executing within a running batch job. The attach service accepts requests to start other Spectrum services and then initiates the services as attached tasks within the batch job.

Action Check the Communication Logs for messages generated by the initiated task. If no messages appear, the task started successfully. Error messages may take some time to appear, depending on how long it takes for the Spectrum service to attempt a startup. Otherwise, you can find the error that caused the Spectrum service to abort in this query.

SPE5096 **Debug filename must be ' ', 'U' or 'T'.**

The Debug Filename field (on the Maintain User Table panel in the Spectrum Administration subsystem) does not contain a valid value. This field must be of type: T (timestamp), U (user ID), or blank (defaults to U). The type determines the naming convention used when the debug file source member is created.

Action To use a timestamp, enter T in the Debug Filename field. To use a user ID, specify U or blank in the field.

- SPE5097** **At least one of Version, Release or SM must be specified.**
- At least one version component must have a non-zero value. As long as one of the values is non-zero, the version is valid (for example, 00/01/00 or 00/00/01).
- Action** Ensure that at least one component (Version, Release, or SM) contains a non-zero value.
- SPE5098** ***Version or release or SM cannot be less than zero.***
- No version component may contain a value less than zero.
- Action** Ensure that no version component (Version, Release, and SM) contains a negative number.
- SPE5099** **Name specified for *Natural module* is invalid.**
- The specified Natural module name is invalid. Natural names follow a set of predefined guidelines.
- Action** Check the guidelines defined for Natural modules in the *Natural 2.2 Reference Manual*.
- SPE5100** ***Field value may not contain embedded blanks.***
- The specified value cannot contain blank characters.
- Action** Check the field value and remove or replace embedded blank characters. If spacing is required, use dashes (-) or underscore characters (_).

SPE5101 ***Service log name must not be specified for service of type service type.***

You cannot specify a service log name for the specified service type.

The Service log name field logically names the Spectrum service, which allows multiple services with different service names to have the same service log name. Since this field determines the browse order for scrolling Service queues, it can also group services with different service names.

Not all services create and maintain service queues. As such, they have no need for this field to be specified.

Action Take the following steps:

- 1 Specify a service type to one that allows a service ID (all service types except Attach).
- 2 Remove the value from the service ID field.

SPE5102 ***Service name found as field ID on service service name.***

You cannot delete a Spectrum service if it is still linked to another Spectrum service through a remote service definition.

Action Take one of the following steps:

- To delete the service, remove all services linked to the service (in the Remote Services field on the first Maintain Services panel) and repeat the request.

Or

- Discontinue the delete action.

- SPE5103** **Invalid DBID; must be in the range *start value* through *end value*.**
- The DBID of the FUSER containing the Debug source member must be within the range of 0 to 254.
- Action** Ensure that the specified value is between 0 and 254.
- SPE5104** **Invalid FNR; must be in the range *start value* through *end value*.**
- The FNR of the FUSER containing the Debug source member must be within the range of 0 to 254.
- Action** Ensure that the specified value is between 0 and 254.
- SPE5105** **Security exit *Natural module* object must exist in Spectrum library *Natural library*.**
- The security exit specified on the Spectrum Control record is not in the Construct Spectrum library.
- Action** If the security exit name is correct, ensure that the object for this module exists in the SYSSPEC library.
- SPE5106** ***Field ID* must have a value of at least 30 (seconds).**
- The specified field value must be 30 or greater.
- Action** Change the field to a value of 30 or more.
- SPE5107** **Invalid parameter value *keyword value*.**
- The service start routine detected an unknown keyword.
- Action** Take the following steps:
- 1 Ensure that the keywords defined in the User parameters string are valid for the service start routine.
 - 2 Check the source for the service start routine and ensure that the list of keywords it handles is correct.

- SPE5108** **Parameter data value is too long keyword value.**
The value associated with the keyword exceeds the maximum parameter length of 100 characters.
- Action** Locate the invalid value and adjust it to fit within a 100 character string.
- SPE5109** **Error at position *byte position* – keyword *keyword* not preceded by *special character*.**
The keyword in the User parameters string is not preceded by a comma or a space.
- Action** Ensure that each keyword is preceded by a valid character.
- SPE5110** **ATTACH= parameter must be a service of type 'A'ttach.**
The value specified for the ATTACH= keyword does not point to a service of type A (attach).
- Action** Ensure that the Spectrum service associated with the ATTACH= parameter is of type A (attach).
- SPE5111** **Spectrum Application Admin secured data is unavailable.**
The secured data for Construct Spectrum cannot be found. Secured data resides in an Adabas file. The file is accessed using logical file number 135.
- Action** Have the Natural administrator:
- 1 Check the current system file settings (using the SYSPROF utility).
 - 2 Check the values of the NTFILE parameters in the specified Natural parameter module.
 - 3 Check the LFILE specifications for file 135 in the specified Natural profile(s).
 - 4 Check the Natural parameters in the batch JCL used for Construct Spectrum.

SPE5112**Spectrum Application Admin unsecured data is unavailable.**

The unsecured data for Construct Spectrum cannot be found. Unsecured data resides in an Adabas file. The file is accessed using logical file number 136.

Action

Have the Natural administrator:

- 1 Check the current system file settings (using the SYSPROF utility).
- 2 Check the values of the NTFILE parameters in the specified Natural parameter module.
- 3 Check the LFILE specifications for file 136 in the specified Natural profile(s).
- 4 Check the Natural parameters in the batch JCL used for Construct Spectrum.

License Key Messages

These error messages may occur when starting the Spectrum Administration subsystem or Spectrum dispatch services. They result from missing or invalid license key information.

- SPE5113** **Current license key expires on *expiry date*.**
- The current copy of Construct Spectrum is a trial copy and has expired. Once the expiry date passes, the system is no longer functional.
- Action** Contact SAGA SOFTWARE to upgrade the trial copy to a licensed copy. As part of this upgrade, a new, permanent license key will be issued. This new license key will not have an expiry date.
- SPE5114** **Licensed to *licensee*.**
- Information only** Confirms the name of the licensee when the license information is entered or updated.
- SPE5115** **Warning: *file version file version does not match code version system version*.**
- The current version of the Spectrum Administration subsystem does not match the version of the data on the Construct Spectrum system files.
- Action** Take the following steps:
- 1 Ensure that the most recent installation updated both the system files and the Spectrum Administration subsystem.
 - 2 Ensure that the FNAT used when running the Spectrum Administration subsystem contains the correct version of Construct Spectrum.
 - 3 Ensure that LFILE=135 and LFILE=136 point to Spectrum files that have the same version as the Spectrum Administration subsystem.

SPE5117 **License key is invalid - contact your local SAGA SOFTWARE office.**

The license key needed for the proper operation of Construct Spectrum is either not found or incorrect.

- Action** Take one or more of the following steps:
- Ensure that you have a license key and that it was entered properly.
 - If you do not have a license key, contact SAGA SOFTWARE support to request one.

SPE5118 **Form printed successfully.**

Information only Indicates that the form containing license key information was successfully routed to a system printer.

SPE5119 ***Field value is required.***

The specified field must contain a value.

- Action** Ensure that the field contains a valid value.

Subprogram Proxy Messages

SPE8450

Incorrect data > *field contents* passed to proxy module name for field > receiving field with format > field format.

The data to be assigned to a target field by the subprogram proxy did not contain a valid value.

Action Take one or more of the following steps:

- Ensure that the data passed to the server matches the data expected. Compare the contents of the target data area with the data definition in the library image file (LIF) defined for the application.
- Ensure that the subprogram proxy was compiled using the correct PDA(s).
- Ensure that the subprogram proxy was generated after any changes to the target module PDA.
- Use the Diagnostics program to retrieve information about each of the data blocks (name, offset, length) and a copy of the initialized data for each block. Ensure that the number of blocks and lengths are as expected. Use the initialized data to locate alignment problems. For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.

SPE8451 **Module *proxy module name* expected *expected bytes* bytes for method *target method*, received *actual bytes* bytes.**

The amount of data received by the subprogram proxy does not match the amount of data expected.

Action Take one or more of the following steps:

- Ensure that the subprogram proxy was compiled using the correct PDA(s).
- Ensure that the subprogram proxy was generated after any changes to the target module PDA.
- Use the Diagnostics program to retrieve information about each of the data blocks (name, offset, length) and a copy of the initialized data for each block. Ensure that the number of blocks and lengths are as expected. Use the initialized data to locate alignment problems. For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.

SPE8452 **Unknown error occurred in program *proxy module name*.**

An error not handled by the ON ERROR clause nor by standard message checking occurred in the specified subprogram proxy.

SPE8453 **Input logic error occurred in *proxy module name*.**

Indicates that the logic that handles blocks within the generated subprogram proxy code is incorrect.

Action Contact SAGA SOFTWARE support.

Natural Messages

These error messages are generated by the Natural nucleus at runtime. They are the most common errors returned by Construct Spectrum's runtime error handlers. Although the Spectrum dispatch service can encounter any Natural runtime error, this section describes the most common ones.

NAT0082 **Invalid command, or object does not exist in library.**

This error is usually caused by a reference to a subprogram proxy (in an application service definition) that does not exist or has not been cataloged.

Action Take the following steps:

- 1 Check the Domain, Object, Version, and Method properties for Request to determine which application service definition is being requested.
- 2 Use the Maintain Application Service Definitions panel (Spectrum Administration subsystem) to check which subprogram proxy is being called for this combination of domain, object, version, and method.
- 3 Ensure that the subprogram is either in the SYSTEM library or cataloged and in the steplib chain associated with this combination of domain, object, version, and method. Use the Spectrum Administration subsystem to determine this information.
- 4 Ensure that no routine with the same name and possibly different interface resides in a higher steplib.

NAT0886 Storage overflow in extended buffer.

For more information, see the documentation for Error 0886 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

- Action** Increase the size of the ESIZE parameter when starting a Spectrum dispatch service in one of the following ways:
- Using the User parameters field on the Maintain Services panel for the Spectrum dispatch service.
 - Using the profile specified for the User parameters field.
 - Dynamically if you are running a Spectrum dispatch service in online mode. (For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.)

NAT0888 Storage overflow during compilation or execution.

For more information, see the documentation for Error 0888 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Action Take one or more of the following steps:

- 1 If the CALLNAT chain is many levels deep, rework the code to reduce the number of levels needed.
- 2 Declare local data areas as PDAs higher up in the CALLNAT sequence, passing the data structure through the CALLNAT sequence and resetting the same space within each subprogram.
- 3 For subprograms deep in the CALLNAT sequence, remove some local data to a global data area used just for the specified subprogram. This causes Natural to use **ESIZE** instead of **DATSIZE** to store this temporary information.
- 4 Increase the size of the **DATSIZE** parameter when starting a Spectrum dispatch service:
 - Using the User parameters field on the Maintain Services panel for the Spectrum dispatch service.
 - Using the profile specified for the User parameters field.
 - Dynamically if you are running a Spectrum dispatch service in online mode. (For more information, see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer's Guide*.)

NAT0935 Conflict in number of parameters.

For more information, see the documentation for Error 0935 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Action Take the following steps:

- 1 Ensure that the path to the target routine is defined correctly on the Steplib table (libraries, DBIDs, and FNRs) and the steplib is associated with the domain, object, or method used to access the target routine.
- 2 Ensure that no target routine with the same name and possibly different interface resides in a higher steplib than where the specified routine resides.

NAT0936 Format/length conflict in parameter.

For more information, see the documentation for error 0936 in the *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Action Take the following steps:

- 1 Ensure that the path to the target routine is defined correctly on the Steplib table (libraries, DBIDs, and FNRs) and the steplib is associated with the domain, object, or method used to access the target routine.
- 2 Ensure that no target routine with the same name and possibly different interface resides in a higher steplib than where the specified routine resides.

NAT0937**Conflict for array definition in parameter.**

For more information, see the documentation for Error 0937 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Action Take the following steps:

- 1 Ensure that the path to the target routine is defined correctly on the Steplib table (libraries, DBIDs, and FNRs) and the steplib is associated with the domain, object, or method used to access the target routine.
- 2 Ensure that no target routine with the same name and possibly different interface resides in a higher steplib than where the specified routine resides.

NAT3017**Invalid file number.**

For more information, see the documentation for Error 3017 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Note: This error may occur if the *LFILE* settings for the two Spectrum Administration subsystem data files are incorrectly defined. Ensure that *LFILE=135* points to the secured data file and *LFILE=136* points to the unsecured data file.

NAT3051**An error in the record buffer was detected.**

For more information, see the documentation for Error 3051 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Note: This error may occur if the *LFILE* settings for the two Spectrum Administration subsystem data files are incorrectly defined. Ensure that *LFILE=135* points to the secured data file and *LFILE=136* points to the unsecured data file.

NAT3055 Attempted format/length conversion not possible.

For more information, see the documentation for Error 3055 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Note: This error may occur if the LFILE settings for the two Spectrum Administration subsystem data files are incorrectly defined. Ensure that LFILE=135 points to the secured data file and LFILE=136 points to the unsecured data file.

NAT3061 An error was detected in the search buffer.

For more information, see the documentation for Error 3061 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Note: This error may occur if the LFILE settings for the two Spectrum Administration subsystem data files are incorrectly defined. Ensure that LFILE=135 points to the secured data file and LFILE=136 points to the unsecured data file.

NAT3148 The database is currently not active.

For more information, see the documentation for Error 3148 in *Natural 2.2 Error Messages* manual or access online help in a Natural session.

Note: This error may occur if the LFILE settings for the two Spectrum Administration subsystem data files are incorrectly defined. Ensure that LFILE=135 points to the secured data file and LFILE=136 points to the unsecured data file.