

**Administrator's Guide**

Version 4.3

**October 1999**

# CONSTRUCT SPECTRUM™

Manual Order Number: **SPE430-040IBM**

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**

|  |    |
|--|----|
| Prerequisites . . . . .                  | 12 |
| How this Guide is Organized . . . . .    | 13 |
| How to Use this Guide . . . . .          | 14 |
| System Administration . . . . .          | 14 |
| Application Administration . . . . .     | 15 |
| Application Development . . . . .        | 15 |
| Conventions Used in this Guide . . . . . | 17 |
| Related Documentation . . . . .          | 19 |
| Construct Spectrum SDK . . . . .         | 19 |
| Construct Spectrum . . . . .             | 20 |
| Natural Construct . . . . .              | 20 |

## **1. OVERVIEW OF THE CONSTRUCT SPECTRUM ADMINISTRATION SUBSYSTEM**

|   |    |
|---|----|
| The Architecture of Construct Spectrum Applications . . . . .         | 22 |
| Mainframe Server . . . . .  | 24 |
| System Functions . . . . .  | 26 |
| Windows . . . . .   | 27 |
| Microsoft IIS (Internet Information Server) . . . . .                 | 30 |
| Internet/Intranet . . . . .   | 31 |
| The Role of the Construct Spectrum Administration Subsystem . . . . . | 32 |
| System Administration . . . . .                                       | 32 |
| Application Administration . . . . .                                  | 33 |
| Security Administration . . . . .                                     | 34 |
| Construct Spectrum's Security Cache . . . . .                         | 35 |
| Using the Construct Spectrum Administration Subsystem . . . . .       | 36 |
| Invoking Construct Spectrum Administration . . . . .                  | 36 |
| Construct Spectrum Administration Menus . . . . .                     | 37 |

|  |    |
|--|----|
| Using Natural Security With Construct Spectrum .....         | 40 |
| Configuring Natural Security for Construct Spectrum .....    | 40 |
| Starting Services in Batch Mode under Natural Security ..... | 41 |
| Natural Security Interface to Restricted Libraries .....     | 41 |
| System Administration Menus and Natural Security .....       | 43 |
| System Administration Maintenance Menu .....                 | 44 |
| System Administration Query Menu .....                       | 44 |
| Using EntireX Security with Construct Spectrum .....         | 45 |

## 2. DEFINING AND MANAGING CONSTRUCT SPECTRUM SERVICES

|   |    |
|---|----|
| Types of Spectrum Services .....                                      | 48 |
| Main and Command Services .....                                       | 49 |
| Defining Construct Spectrum Services .....                            | 51 |
| Accessing the Maintain Services Panels .....                          | 51 |
| Overview of the Maintain Services Panels .....                        | 52 |
| Identifying the Spectrum Service .....                                | 54 |
| Defining Entire Broker Service Settings .....                         | 54 |
| Server Names Supplied with Construct Spectrum .....                   | 57 |
| Attach Servers .....  | 57 |
| Linking Services to an Attach Server .....                            | 57 |
| Activation Policies .....   | 58 |
| Runtime Error Handling .....  | 58 |
| Common Server Startup Interface .....                                 | 58 |
| Using the START Command to Initiate Services as Subtasks .....        | 59 |
| Using the START Command to Initiate Services Online or in Batch ..... | 59 |
| Defining Service Start Parameters and Service Start Routines .....    | 59 |
| Using Service Start Routines .....                                    | 60 |
| BATCHJOB Service Start Routine .....                                  | 61 |
| Example BATCHJOB Routine .....  | 63 |
| SUBTASKB Service Start Routine .....                                  | 64 |
| Example SUBTASKB Routine .....  | 64 |
| Writing to Print and Workfiles .....                                  | 65 |
| Generating Trace Output .....   | 65 |
| Defining Spectrum Service Settings and Program Parameters .....       | 67 |
| Changing the Attribute File .....                                     | 70 |
| Adding a Natural Profile .....  | 71 |

|  |    |
|--|----|
| Managing Construct Spectrum Services . . . . .             | 72 |
| Accessing the Manage Services Panel . . . . .              | 73 |
| Using the Manage Services Panel . . . . .                  | 73 |
| Overview of Commands . . . . .                             | 75 |
| Debug Command . . . . .                                    | 77 |
| How the Debug Command Works . . . . .                      | 77 |
| EndJob Command . . . . .                                   | 77 |
| EndService Command . . . . .                               | 78 |
| EnVironment Command . . . . .                              | 78 |
| Initiate Command . . . . .                                 | 78 |
| Ping Command . . . . .                                     | 80 |
| Refresh Command . . . . .                                  | 80 |
| Shutdown Command . . . . .                                 | 81 |
| Shutting Down a Single Replica . . . . .                   | 81 |
| Shutting Down All Replicas of a Service . . . . .          | 81 |
| Shutting Down All Servers Within a Batch Job . . . . .     | 81 |
| Example of Shutting Down Servers in a Batch Job . . . . .  | 81 |
| STeplibs Command . . . . .                                 | 83 |
| SYsprof Command . . . . .                                  | 83 |
| Standard Response Protocol . . . . .                       | 83 |
| Accessing the Query Communications Logs Panel . . . . .    | 83 |
| Queries and Logs for Construct Spectrum Services . . . . . | 84 |
| Accessing Query Panels . . . . .                           | 85 |
| Accessing Query Log Panels . . . . .                       | 87 |

### **3. USING THE CONSTRUCT SPECTRUM CONTROL RECORD**

|  |    |
|--|----|
| Accessing the Maintain Control Record Window . . . . . | 90 |
| Using the Maintain Control Record Window . . . . .     | 91 |
| Modifying Control Data . . . . .                       | 92 |
| Natural Security . . . . .                             | 92 |
| Security Exit . . . . .                                | 92 |
| Entire Net-Work Timeout . . . . .                      | 93 |
| Spectrum Control Record Log . . . . .                  | 94 |
| Accessing the Spectrum Control Record Log . . . . .    | 94 |

#### 4. DEFINING GROUPS AND USERS

|   |     |
|---|-----|
| Introduction  | 96  |
| Defining Groups                                       | 99  |
| Defining Groups Using Spectrum Security               | 99  |
| Accessing the Maintain Group Table Panel              | 99  |
| Defining Groups Using Natural Security                | 100 |
| Accessing the Maintain Group Profile Window           | 101 |
| Using the Maintain Group Profile Window               | 102 |
| Defining Users  | 104 |
| Defining Users Using Spectrum Security                | 104 |
| Accessing the Maintain User Table Panel               | 104 |
| Defining a Debug Library and File                     | 105 |
| Associating Users with Groups                         | 106 |
| Defining Users Using Natural Security                 | 107 |
| Accessing the Maintain User Profile Window            | 107 |
| Using the Maintain User Profile Window                | 108 |
| Associating Users with Groups                         | 109 |
| Accessing the Maintain User's Groups Window           | 110 |
| Linking Users to Libraries                            | 111 |
| Accessing the Library Maintenance / Link Users Window | 111 |
| Using the Library Maintenance / Link Users Window     | 112 |
| Queries and Logs for User and Group Tables            | 113 |
| Queries and Logs for Spectrum Security                | 113 |
| Accessing Query Panels                                | 114 |
| Accessing Query Log Panels                            | 116 |
| Queries for Natural Security                          | 117 |
| Accessing Query Panels                                | 117 |
| Query Group Table                                     | 118 |
| Query User Table                                      | 119 |
| Query User's Groups                                   | 120 |

#### 5. SETTING CONSTRUCT SPECTRUM SECURITY OPTIONS

|   |     |
|---|-----|
| Introduction                                  | 124 |
| Defining Security Settings                    | 125 |
| Accessing the Maintain Object Security Panels | 125 |
| Adding a Group and Domain Combination         | 126 |

|   |     |
|---|-----|
| Purging a Group-Domain Combination .....                    | 127 |
| Setting Security Options .....                              | 127 |
| Setting Security Options for Group-Domain Combinations..... | 127 |
| Setting Security Options for Objects.....                   | 128 |
| Setting Security Options for Methods.....                   | 130 |
| Security Statuses .....                                     | 131 |
| No Privileges Status.....                                   | 131 |
| Revoked and Granted Statuses .....                          | 132 |
| Disabled and Enabled Statuses .....                         | 132 |
| Using the Security Exit Modules.....                        | 133 |
| BS_USEC0 (Template).....                                    | 133 |
| BS_USEC1 (“Pessimistic” Checking) .....                     | 134 |
| BS_USEC2 (“Optimistic” Checking) .....                      | 135 |
| Using the Cache Synchronization Exit Module .....           | 136 |
| Queries and Logs for Spectrum Security .....                | 137 |
| Accessing Query and Log Panels.....                         | 137 |

## 6. USING CONSTRUCT SPECTRUM TOOLS

|  |     |
|--|-----|
| System Administration Tools .....                                  | 140 |
| Accessing the System Administration Tools.....                     | 140 |
| Synchronizing the Security Cache and Cleanup Queue.....            | 141 |
| Security Cache .....   | 141 |
| Cleanup Queue .....  | 141 |
| Using the Synchronize Security Cache & Cleanup Queue Function..... | 142 |
| Resetting the Security Cache.....                                  | 142 |
| Client Tools.....  | 143 |
| Spectrum Service Manager .....                                     | 143 |
| Service Properties Dialog.....                                     | 145 |
| Change Password Program .....                                      | 149 |
| Remote Clock Program.....  | 151 |
| Using the TIMESTAMP Service .....                                  | 151 |
| Using the Spectrum Dispatch Client.....                            | 152 |
| Command Line Parameters.....                                       | 152 |
| Broker Driver Program .....  | 154 |

## 7. DEPLOYING THE CONSTRUCT SPECTRUM ADMINISTRATION SUBSYSTEM

|  |     |
|--|-----|
| Introduction   | 158 |
| Example Deployment Scenario  | 159 |
| Creating the Example Deployment Scenario   | 161 |
| Step 1 — Ensure that the Production Environments have the Prerequisite Software Installed      | 161 |
| Step 2 — Ensure that Construct Spectrum is Installed on Each Machine                           | 161 |
| Step 3 — Copy Construct Spectrum System Files  | 162 |
| Logical File 135   | 162 |
| Logical File 136   | 162 |
| Step 4 — Ensure that the Spectrum System File Database is set up in Entire Net-Work (Optional) | 162 |
| Step 5 — Define and Start an Attach Service  | 163 |
| Step 6 — Define and Start Dispatch and Security Services                                       | 163 |
| Natural Security   | 163 |
| Step 7 — Transfer Domains  | 164 |
| Step 8 — Transfer Group and User Tables (Optional)   | 164 |
| Data Transfer Utilities  | 165 |
| Transferring Domains   | 165 |
| Accessing the Domain Transfer Utilities  | 166 |
| Exporting Domains  | 167 |
| Importing Domains  | 168 |
| Transferring Groups  | 169 |
| Accessing the Group Transfer Utilities   | 169 |
| Exporting Groups   | 171 |
| Importing Groups   | 172 |
| Using the Data Transfer Utilities in Batch Mode  | 173 |
| Exporting Domains  | 173 |
| Importing Domains  | 173 |
| Exporting Groups   | 174 |
| Importing Groups   | 174 |
| Layout of the Export Users and Groups Workfile   | 175 |
| The Header Record  | 176 |
| The Group Record   | 176 |
| The User Record  | 177 |

**APPENDIX A — USING BSSPARMN TO PARSE USER  
PARAMETERS..... 179**

    BSSPARMA ..... 179

    CDPDA-M..... 180

**APPENDIX B — USING SPSERRN FOR ADDITIONAL ERROR-  
HANDLING..... 183**

**INDEX ..... 185**



## PREFACE

Welcome to the Construct Spectrum Administration subsystem — your tool for setting up and managing environments for Construct Spectrum applications. The *Construct Spectrum Administrator's Guide* is for administrators who use the subsystem to manage network communications and applications. This preface will help you get the most out of the guide and find other sources of information about managing Construct Spectrum applications.

The following topics are covered:

- **Prerequisites**, page 12
- **How this Guide is Organized**, page 13
- **How to Use this Guide**, page 14
- **Conventions Used in this Guide**, page 17
- **Related Documentation**, page 19

## Prerequisites

The *Construct Spectrum Administrator's Guide* does not provide information about the following topics. We assume that you are either familiar with these topics or have access to other sources of information about them.

- Natural programming language and environment
- Natural Construct
- Entire Broker
- Entire Net-Work

See **Related Documentation**, page 19 for sources of information about Natural Construct and Construct Spectrum.

---

## How this Guide is Organized

This section explains the contents of each chapter in the *Construct Spectrum Administrator's Guide*. For information about how to use the guide depending on the type of action you wish to perform, see **How to Use this Guide**, page 14.

This guide contains the following chapters:

- 1 **Overview of the Construct Spectrum Administration Subsystem**  
This chapter introduces the architecture of a Construct Spectrum client/server and web applications and the role of the Construct Spectrum Administration subsystem in supporting applications. Also turn to this chapter for information about accessing and navigating the subsystem.
- 2 **Defining and Managing Construct Spectrum Services**  
This chapter explains how to create and manage Spectrum services that automate communication between application components and Entire Broker.
- 3 **Using the Construct Spectrum Control Record**  
This chapter explains how to use the Construct Spectrum control record to initiate, ping, and shutdown services, as well as how to enable Natural Security, specify a security exit, and adjust Entire Net-Work timeouts.
- 4 **Defining Groups and Users**  
This chapter explains how to define groups, users, and their associations. Groups and users are essential for creating a secure environment, whether you are using Construct Spectrum security or Natural Security.
- 5 **Defining Construct Spectrum security Options**  
This chapter explains how to define Construct Spectrum security settings for domains, objects, and methods.
- 6 **Using Construct Spectrum Tools**  
This chapter explains the tools on the client and server for managing the environment of Construct Spectrum applications.
- 7 **Deploying the Construct Spectrum Administration Subsystem**  
This chapter explains the steps involved in deploying the subsystem to run applications in production environments.
- A **Appendix: Using the BSSPARMN Utility to Parse User Parameters**  
This appendix explains how to use the BSSPARMN to parse the keywords specified in the Service Start parameter fields in the Maintain Services panel.
- B **Appendix: Using SPSEERRN for Additional Error-Handling**  
This appendix explains how to use this user exit to enhance core error-processing in Construct Spectrum applications.

## How to Use this Guide

The Construct Spectrum Administration subsystem is used for three types of activities:

- system administration
- application administration
- application development.

The following sections explain how to use this and related Construct Spectrum guides to perform the three types of tasks.

We recommend that anyone using the Construct Spectrum Administration subsystem read **Overview of the Construct Spectrum Administration Subsystem**, page 21, to learn how to access and navigate the subsystem and to gain a general understanding of the subsystem's role in supporting Construct Spectrum applications.

## System Administration

- The activities of a system administrator include setting up and deploying the Construct Spectrum Administration subsystem.
  - For information about installing and setting up the Construct Spectrum Administration subsystem, see the *Construct Spectrum and SDK Client Installation Guide*.
  - For information about deploying the Construct Spectrum Administration subsystem in one or more production or development environments, see **Deploying the Construct Spectrum Administration Subsystem**, page 157.
- As a system administrator, you will use the Spectrum Administration subsystem to set up and maintain Spectrum services. Spectrum services interact with Entire Broker and your network to automate the communication between application components on the client and server. They also handle a Spectrum application's security checking, whether you are using Construct Spectrum security or Natural Security.
  - See **Defining and Managing Construct Spectrum Services**, page 47.

- 
- You will use the Construct Spectrum control record to adjust Entire Net-Work timeouts, enable or disable Natural Security, and specify a security exit, if required.
    - See **Using the Construct Spectrum Control Record**, page 89.
  - You will also need to define users and groups to the Construct Spectrum Administration subsystem. The application administrator can then define the groups' access privileges to application domains, objects, and methods.
    - See **Defining Groups and Users**, page 95.
  - You will also be involved in monitoring and adjusting the performance of the Construct Spectrum Administration subsystem.
    - For information about synchronizing the security cache and cleanup queue and resetting the security cache, see **Using Construct Spectrum Tools**, page 139.
    - For information about resolving errors, see *Construct Spectrum Messages*.

## Application Administration

- As an application administrator, you will access the Spectrum Administration subsystem to maintain the security of applications. This involves specifying groups' access privileges to domains, objects, and methods.
  - See **Setting Construct Spectrum Security Options**, page 123.
- You may also wish to access the Construct Spectrum Administration subsystem to maintain application service definitions.
  - See **Using the Subprogram Proxy Model**, page 129, *Construct Spectrum Programmer's Guide*.

## Application Development

- In the course of developing a Construct Spectrum application, you will access the Construct Spectrum Administration subsystem to define a steplib chain and domain.
  - See **Setting up Your Application Environment on the Mainframe**, page 45, *Construct Spectrum Programmer's Guide*.

- You may also have to set security options for the development environment so that you can test the application.
  - For information about defining groups and users, see **Defining Groups and Users**, page 95.
  - For information about linking groups to domains, see **Setting Construct Spectrum Security Options**, page 123.
- You may also wish to access the Construct Spectrum Administration subsystem to maintain application service definitions.
  - See **Using the Subprogram Proxy Model**, page 129, *Construct Spectrum Programmer's Guide*.

## Conventions Used in this Guide

This guide uses the following typographical conventions:

| <b>Example</b>   | <b>Description</b>  |
|--|---|
| <b>Introduction</b>  | Bold text in cross references indicates chapter and section titles.   |
| “A”  | Quotation marks indicate values you must enter.   |
| Browse model,<br>GotFocus, Enter                                       | Mixed case text indicates: <ul style="list-style-type: none"> <li>• The names of Natural Construct and Construct Spectrum editors, fields, files, functions, models, panels, parameters, subsystems, variables, and dialogs.</li> <li>• The names of Visual Basic classes, constants, controls, dialogs, events, files, menus, methods, properties, and variables.</li> <li>• The names of keys.</li> </ul> |
| 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<br/>Administrator’s Guide,<br/>variable name</i> | Italicized text indicates: <ul style="list-style-type: none"> <li>• Book titles.</li> <li>• Placeholders for information you must supply.</li> </ul>  |



| <b>Example</b>               | <b>Description (continued)</b>  |
|------------------------------|---|
| <code>[variable]</code>      | In syntax and code examples, values within square brackets indicate optional items.   |
| <code>{WHILE   UNTIL}</code> | 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 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 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).
- *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 Reference Manual*  
This manual is for application developers and administrators who need quick access to information about Construct Spectrum application programming interfaces (APIs) and utilities.
- *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.

## 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.

---

# OVERVIEW OF THE CONSTRUCT SPECTRUM ADMINISTRATION SUBSYSTEM

Reading this chapter will give you a general understanding of the architecture of a Construct Spectrum application and the role that the Construct Spectrum Administration subsystem plays in supporting the application. You will also learn how to access and navigate the subsystem and how to use Natural Security with the subsystem.

The following topics are covered:

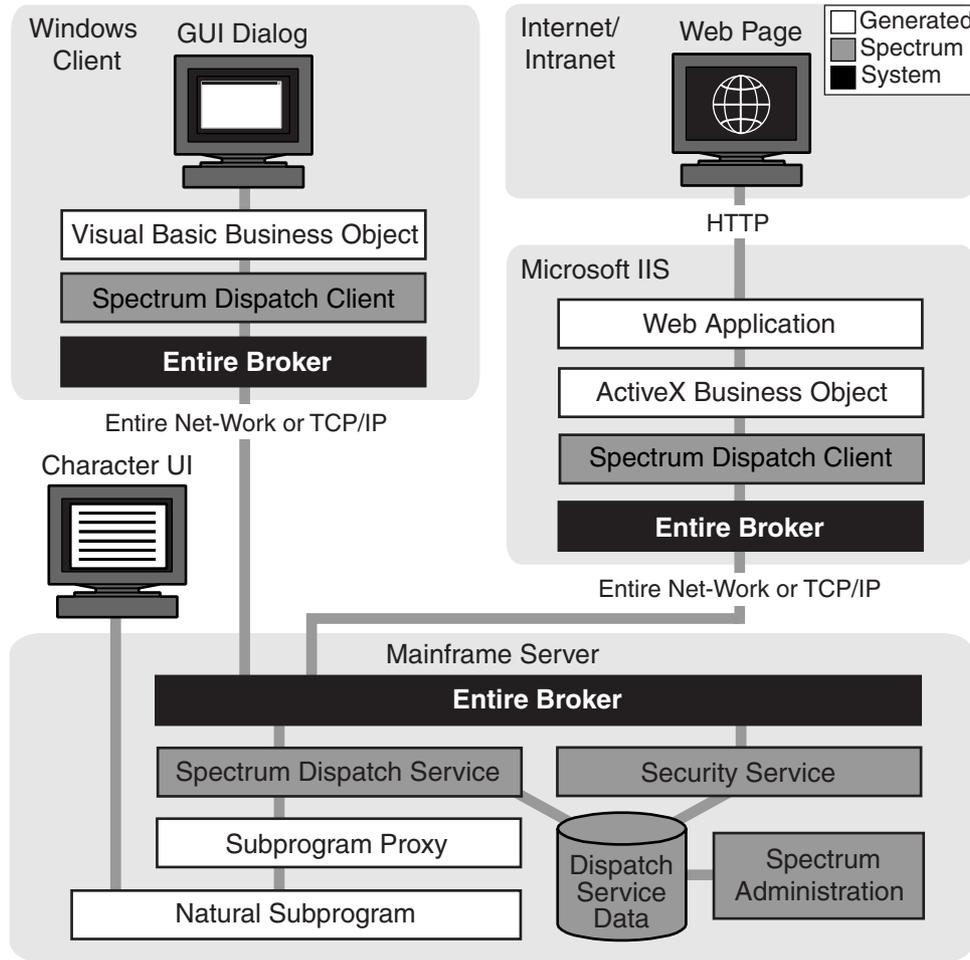
- **The Architecture of Construct Spectrum Applications**, page 22
- **The Role of the Construct Spectrum Administration Subsystem**, page 32
- **Using the Construct Spectrum Administration Subsystem**, page 36
- **Using Natural Security With Construct Spectrum**, page 40
- **Using EntireX Security with Construct Spectrum**, page 45

## The Architecture of Construct Spectrum Applications

Using Construct Spectrum and the software development kit (SDK), application developers can create all components of client/server and web application, including Natural object subprograms that perform maintenance and browse functions, and GUI dialogs or web pages that communicate with these Natural object subprograms.

Communication between server and client components of an application is performed by a combination of Entire Broker and Entire Net-Work (or Entire Broker configured to use TCP/IP), as well as Construct Spectrum's middleware components: the Spectrum Dispatch Client and Spectrum services. The Construct Spectrum middleware components encapsulate calls to Entire Broker on the client and server.

The following diagram shows the architecture of Natural character-based applications, client/server applications, and web applications.



Architecture of Construct Spectrum Applications

The following sections explain these components according to the platforms on which the components run: mainframe server, Windows, Microsoft IIS, and internet or intranet.

## Mainframe Server

| Component          | Description   |
|--------------------|---|
| Natural subprogram | Natural subprograms perform maintenance and browse functions on the mainframe server. The same set of business objects can be accessed from character-based Natural applications, client/server applications, and web applications. This ensures that the integrity of business data is preserved, independent of the presentation layer.   |
| Character UI       | Non-distributed Natural applications created with Natural Construct accessing subprograms directly.   |
| Subprogram proxy   | <p>A subprogram proxy acts as a bridge between a specific subprogram and the Spectrum dispatch service. The subprogram proxy:</p> <ul style="list-style-type: none"> <li>• provides a common interface so that the Spectrum dispatch service can pass the same set of parameters to any subprogram proxy</li> <li>• issues a CALLNAT to the subprogram</li> <li>• converts the parameter data of the subprogram into a format that can be transmitted between the client and server</li> <li>• supports optimization of the data passed through the network so that only input parameters need to be sent to the Spectrum dispatch service and only output parameters need to be returned to the client</li> <li>• validates the format and length of the data received from the client</li> <li>• supports debugging features to help uncover inconsistencies between the data sent by the client and the data expected by the subprogram proxy</li> </ul> |

For more information, see **Using the Subprogram Proxy Model**, page 129, *Construct Spectrum Programmer's Guide*.

| Component                 | Description (continued)   |
|---------------------------|---|
| Spectrum dispatch service | <p>The Spectrum dispatch service provides a common interface and Entire Broker services for Natural subprograms in the application. The main functions of the Spectrum dispatch service are to:</p> <ul style="list-style-type: none"><li>• receive requests from the client by way of Entire Broker</li><li>• optionally decompress or decrypt (or both) and translate the request message (see <b>System Functions</b>, page 26) from the client's character set (ASCII) to the server's character set (either ASCII or EBCDIC)</li><li>• check Security to ensure that the client is allowed to issue such a request</li><li>• determine the name of the subprogram proxy that handles the request</li><li>• issue a CALLNAT to the subprogram proxy, passing the received message as a parameter string</li><li>• optionally compress, encrypt (or both) the message to be returned (see <b>System Functions</b>, page 26)</li><li>• send information received from the subprogram proxy back to the client application</li></ul> |
| Dispatch service data     | <p>The information defined and maintained in the Spectrum Administration subsystem is accessed by Spectrum dispatch services anywhere on the network by way of Entire Broker.</p>   |
| Spectrum administration   | <p>This mainframe subsystem allows system administrators, application administrators, and application developers to set up and manage system and application environments.</p>  |

| <b>Component</b> | <b>Description (continued)</b>   |
|------------------|--|
| Security service | <p>A Spectrum security service checks client requests against the security settings defined in the Construct Spectrum Administration subsystem. This stand-alone service operates independently of any one Spectrum dispatch service. Its independence allows the security service to process, in one central location, the requests of several Spectrum dispatch services, which may be located on nodes throughout the network.</p> <p>For more information about security services and security settings, see <b>Setting Construct Spectrum Security Options</b>, page 123.</p> |
| Entire Broker    | Entire Broker transfers messages between Windows or the web server and the Natural environment. Entire Broker can be configured to use either native TCP/IP or Entire Net-Work as the transport layer.   |

## System Functions

All Spectrum dispatch services defined in the Construct Spectrum Administration subsystem have access to common system functions:

| <b>Function</b>              | <b>Description</b>  |
|------------------------------|---|
| Return debugging information | Ensures that all requested debugging information is generated into the source area. Debugging information is requested by setting a Trace-Option in the subprogram proxy. The debugging information is stored as a source member that can be examined or used to initiate the request locally on the server, removing the client and the network from the test. |
| Encrypt and decrypt data     | Supplies an interface that can be called by the external (assembler or C) routines used to encrypt and decrypt data.  |

| <b>Function</b>              | <b>Description (continued)</b>   |
|------------------------------|--|
| Compress and decompress data | Supplies an interface that can be called by the external (assembler or C) routines used to compress and decompress data.   |
| Error handling               | Manages the capturing of runtime errors, returning the errors to the client. If possible, this function also restarts the service that ended with the runtime error.               |
| Message handling             | Returns a message string based on a message number and substitution values. The function accepts and updates the data used by the Spectrum dispatch service to return the message. |
| Data translation             | Translates data received from the client into EBCDIC or ASCII, depending on the requirements of the server.  |

## Windows

Construct Spectrum client/server applications run on Windows or Windows NT.

| <b>Component</b>               | <b>Description</b>   |
|--------------------------------|--|
| Entire Broker                  | Entire Broker transfers messages between the client and the Natural environment. Entire Broker can be configured to use either native TCP/IP or Entire Net-Work as the transport layer.  |
| Spectrum Dispatch Client (SDC) | This Component Object Model (COM) middleware component enables Construct Spectrum applications to read from, and write to, variables in a Natural parameter data area (PDA) and to issue CALLNAT statements to Natural subprograms.<br><br>The main functions of the Spectrum Dispatch Client are: |

| Component | Description (continued)   |
|-----------|---|
|           | <ul style="list-style-type: none"><li data-bbox="635 310 1370 575">• <b>Natural parameter data area simulation</b><br/>The Spectrum Dispatch Client defines the parameter data of Natural business objects as a series of Natural data fields, which may include structures, arrays, and redefines. To call a business object, the Construct Spectrum application must be able to assign values to these parameter data fields before calling the business object and then read the fields after the data is returned from the server.<br/><br/>To facilitate this, the Spectrum Dispatch Client simulates Natural parameter data areas, allowing the application developer to create code that allocates a data area and reads and writes the fields in the data area. Natural parameter data areas residing in a library on the server may be downloaded (by the Construct Spectrum Add-In) to the client. This lets the Spectrum Dispatch Client know the structure (field names and formats) of a parameter data area. Parameter data areas are stored in the library image file on the client and only need to be downloaded after creation or whenever they change on the server.</li><li data-bbox="635 969 1370 1175">• <b>CALLNAT simulation</b><br/>The Spectrum Dispatch Client allows a Construct Spectrum application to issue a CALLNAT to a Natural subprogram. All that needs to be specified in the client code is the logical name of the subprogram to be called and the list of parameter data areas to pass to the subprogram.</li><li data-bbox="635 1193 1370 1338">• <b>Encapsulation of Entire Broker calls</b><br/>The Spectrum Dispatch Client uses Entire Broker calls to communicate with the Spectrum dispatch service. These calls are not exposed within the application layer, so the developer never needs to code Entire Broker calls.</li></ul> |

| Component                    | Description (continued)  |
|------------------------------|--|
|                              | <ul style="list-style-type: none"><li data-bbox="635 306 1366 605">• Database transaction control<br/>Often, two or more calls to subprograms occur within the same database transaction such that an END TRANSACTION statement can be issued if all calls complete successfully. Also, it is advantageous to have the client application control the point at which the END TRANSACTION or BACKOUT TRANSACTION statement occurs. The Spectrum Dispatch Client and the Spectrum dispatch service cooperate to provide these capabilities.</li></ul> <p data-bbox="635 628 1366 718">For more information, see <b>Understanding The Spectrum Dispatch Client</b>, page 243, <i>Construct Spectrum Programmer's Guide</i>.</p> |
| Visual Basic business object | A Visual Basic business object is a Visual Basic class that acts as an intermediary between a dialog and the Spectrum Dispatch Client. This class invokes the methods of subprograms on behalf of dialogs and instantiates all the data areas required to communicate with a subprogram. Visual Basic business objects can also perform local data validation to provide immediate feedback to the user without involving a network call.  |
| GUI dialog                   | GUI dialogs represent graphical interface screens that communicate with the user and interact with the Visual Basic business objects and other framework components to implement business processes.   |

## Microsoft IIS (Internet Information Server)

Web applications created with Construct Spectrum work with Microsoft Internet Information Server (IIS).

| Component                | Description  |
|--------------------------|--|
| Entire Broker            | Entire Broker transfers messages between the web server and the Natural environment. Entire Broker can be configured to use either native TCP/IP or Entire Net-Work as the transport layer.  |
| Spectrum Dispatch Client | <p>This Component Object Model (COM) middleware component enables web applications to read from, and write to, variables in a Natural parameter data area (PDA) and to issue CALLNAT statements to Natural subprograms. Its main functions are simulating PDAs and CALLNATs, encapsulating Entire Broker calls, and controlling database transactions. As the client counterpart of Spectrum dispatch services, it is also responsible for such things as data marshaling, encryption, compression, error-handling, and all Entire Broker communication.</p> <p>For more information, see <b>Understanding The Spectrum Dispatch Client</b>, page 243, <i>Construct Spectrum Programmer's Guide</i>.</p> |
| ActiveX Business Object  | Each back-end business object is represented on the web server as an ActiveX object. This object encapsulates all of the communication with the Spectrum Dispatch Client, making it efficient to invoke Natural services from the client.  |
| Web application          | A Construct Spectrum web application consists of framework components supplied with all Construct Spectrum web projects and components that you generate using Construct Spectrum wizards. Generated components are HTML templates, page handlers, and object factory entries.   |

---

## **Internet/Intranet**

Construct Spectrum web applications support Internet Explorer and Netscape Navigator browsers at version 4.0 and later.

## The Role of the Construct Spectrum Administration Subsystem

The Construct Spectrum Administration subsystem consists of a table-driven server environment based in Natural. The subsystem is used for three types of activity: system administration, application administration, and security administration.

### System Administration

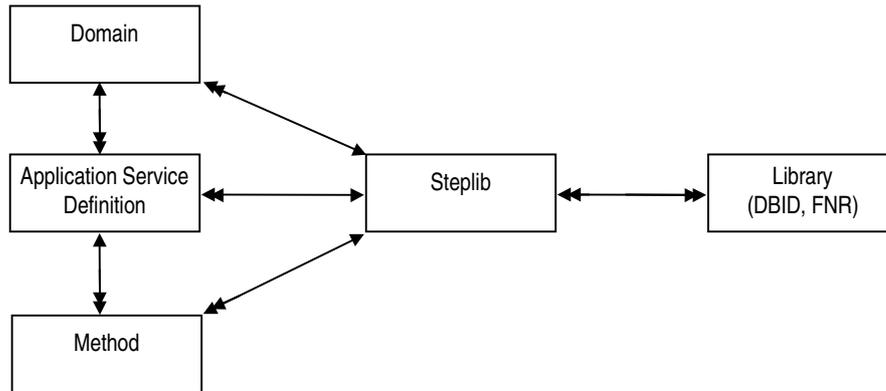
The Construct Spectrum Administration subsystem provides dynamic, online management of tables used by Spectrum services. All data needed by services to start, stop, and perform requested functions is maintained here. The subsystem also provides mechanisms to enhance the functionality of Entire Broker, such as a security mechanism, service start-up and shut-down mechanisms, and messaging mechanisms. It is also used to expose some of Entire Broker's parameters through table-driven maintenance functions, including defining Entire Broker service settings.

System administration involves the following tasks:

- **Defining Spectrum services**  
Spectrum services interact with Entire Broker and the network to automate the communication between application components on the client and server. Spectrum services include attach, dispatch, and security services. Service definitions enable the services to make themselves known to Entire Broker and to configure themselves for the environments on which they are expected to run.
- **Managing Spectrum services**  
Using the Construct Spectrum Administration subsystem, administrators can start, stop, and check ("ping") Spectrum services.
- **Viewing information logs**  
The logs maintained by the Construct Spectrum Administration subsystem contain updates to all system tables, information generated by running the Spectrum services, errors related to communication with the Spectrum dispatch service, and security violations.

## Application Administration

Another way of looking at the architecture of a Construct Spectrum application is to view the components of the Construct Spectrum Administration subsystem that participate in directing requests from the client to the appropriate method on the server. The following diagram illustrates the relationships between components in the Construct Spectrum Administration subsystem:



Application Administration Architecture

The following table explains the components.

| Component           | Description   |
|---------------------|---|
| Steplib and library | A steplib defines a set of Natural libraries that a Spectrum dispatch service must access to call a subprogram proxy. The dispatch service accesses the libraries in the order they are listed in the steplib definition. |

| Component                                 | Description (continued)   |
|---|---|
| Domain                                    | A domain is an identifier used to group business objects. Often, a domain defines the boundaries of an application. For example, all the objects (such as Customer, Order, and Product) in an Order Entry application might share a single domain.  |
| Application service definition and method | An application service definition groups a domain, a business object, and version information under one identifier that describes the business object. The application service definition specifies all the methods of a business object, which subprogram proxies enable access to specific methods, and where the subprograms are located (steplibs). |

Application administrators and developers use the Construct Spectrum Administration subsystem to define steplibs and domains. Application service definitions are also stored in the subsystem, where they can be modified as needed. Usually, developers create application service definitions by using a super model or Subprogram Proxy model to generate application components.

## Security Administration

The Construct Spectrum Administration subsystem is used to define security settings that control users' access to domains, business objects, and methods.

Users' IDs are defined to the Construct Spectrum Administration subsystem. Each user is then assigned to one or more groups. Users and groups can also be defined in Natural Security, rather than Spectrum security.

Groups provide a mechanism for identifying users who need to access the same combination of domains, business objects, and methods. Groups usually associate people who perform similar tasks. For example, an organization might have groups for Quality Assurance, Sales, and Management.

Once users and groups are defined in the Construct Spectrum Administration subsystem, security settings can be defined at multiple levels, as shown in the following table:

| <b>Level</b> | <b>Description</b>  |
|--------------|---|
| Domain       | When a group is granted access to a domain, its members can use all objects and methods in the domain.                    |
| Object       | Within domains, individual objects can have security settings defined for them so that users can access selected objects. |
| Method       | Access privileges can also be explicitly granted or revoked for individual methods of an object.                          |

### **Construct Spectrum's Security Cache**

Because security data is located in the Construct Spectrum Administration subsystem, it can be maintained centrally. In addition, the subsystem includes a dynamically updated cache mechanism that improves throughput. When a Spectrum dispatch service receives a request from the client, the Spectrum security service assigned to it checks the user's security privileges. The Spectrum security service builds a security cache when it first checks the security for a given request. Thereafter, only one table lookup is required to check the access privileges of a user to requested data, thereby improving response time after the initial request.

The contents of the security cache are always dynamic. If changes occur in the Construct Spectrum Administration subsystem data that affect security, the security cache is automatically updated. The security cache is maintained as a separate encrypted table on an external data source.

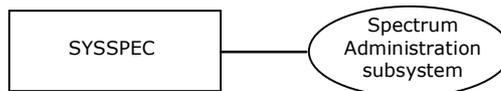
## Using the Construct Spectrum Administration Subsystem

This section explains how to invoke the Construct Spectrum Administration subsystem. It also illustrates the structure of the menu system.

For general information about navigating menus, operating panels, and using on-line help, direct commands, and PF-keys, see **Introduction to Natural Construct**, page 39, *Natural Construct Generation User's Manual*.

### Invoking Construct Spectrum Administration

The Construct Spectrum Administration subsystem resides in the SYSSPEC library:



Construct Spectrum Library

- To invoke the Construct Spectrum Administration subsystem:
  - Enter “Menu” at the NEXT prompt.  
The Construct Spectrum Administration Subsystem Main Menu is displayed.

For more information, see the next section, **Construct Spectrum Administration Menus**.

## Construct Spectrum Administration Menus

When you invoke the Construct Spectrum Administration subsystem, the following menu is displayed:

```

BS__MAIN          Construct Spectrum Administration Subsystem          CDLAYMN1
Mar 11            Main Menu                                           08:55 AM

                Functions
                -----
                SA   System Administration
                AA   Application Administration

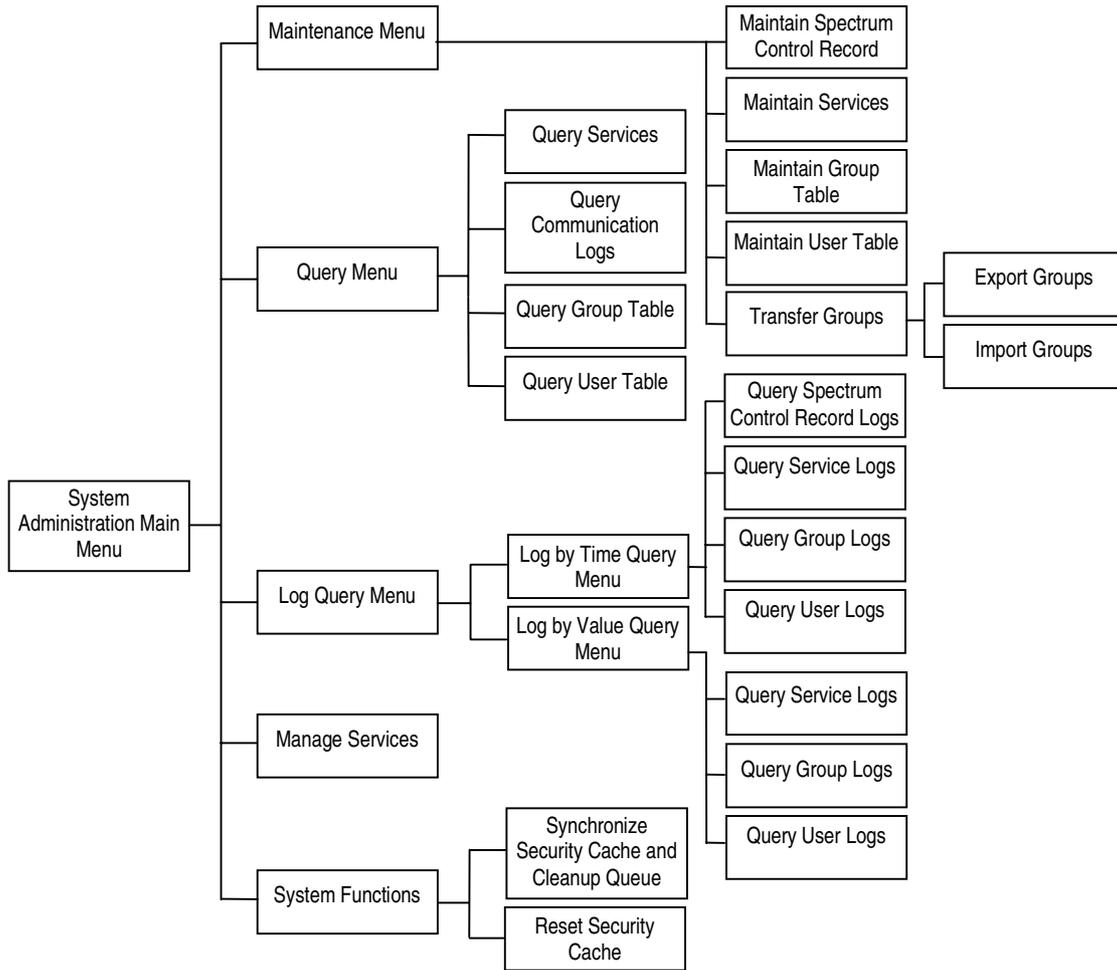
                ?   Help
                .   Terminate
                -----
Function .....  _

Command .....  _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--
      help  retrn quit          flip                                main
    
```

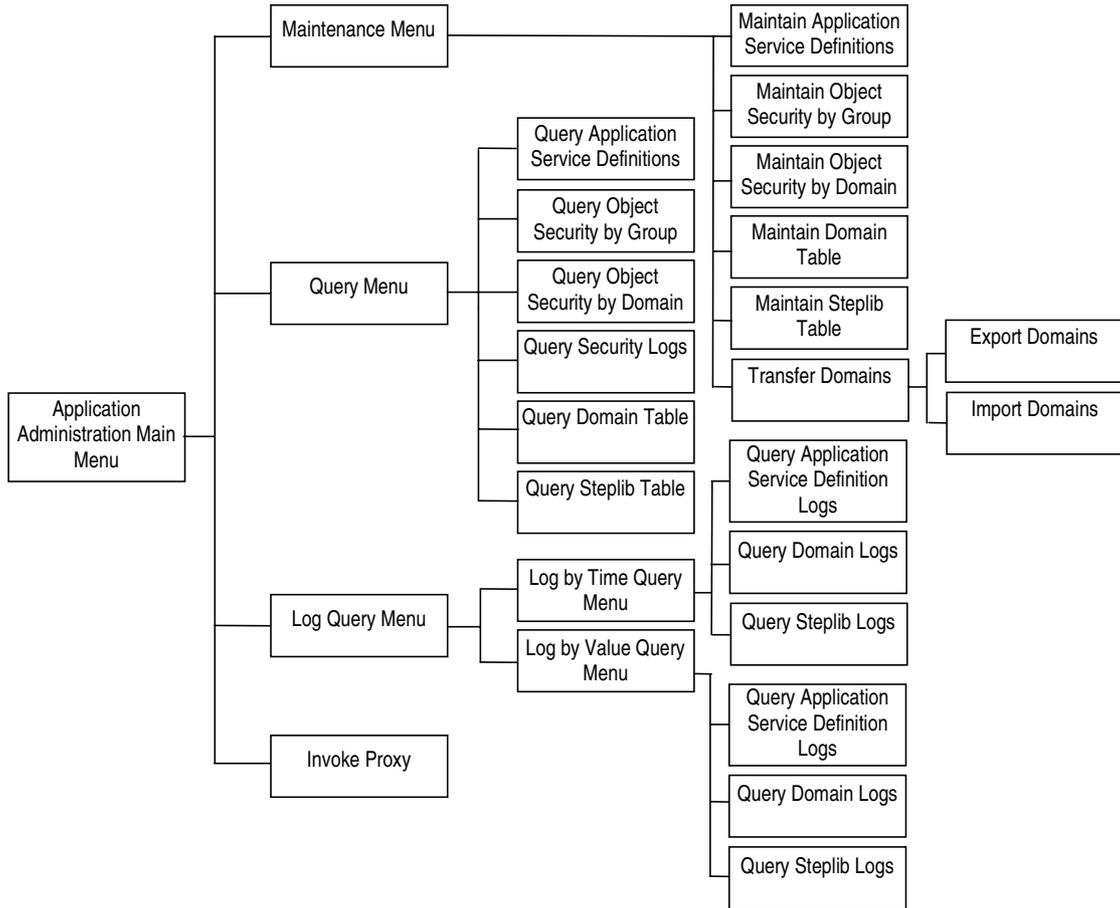
### Construct Spectrum Administration Subsystem Main Menu

The Construct Spectrum Administration subsystem is divided into System Administration and Application Administration functions. Each branch has its own main menu which leads to maintenance, query, and special functions. The following diagrams show the menu structures.

*Note: If Natural Security is enabled, different options are displayed on the System Administration menus. For more information, see **Using Natural Security With Construct Spectrum**, page 40.*



System Administration Menu Structure



Application Administration Menu Structure

## Using Natural Security With Construct Spectrum

The Construct Spectrum subsystem is fully integrated with Natural Security. If Natural Security is being used, all updates to group and user information are applied directly to Natural Security data. It is also possible to define user-library links. If Natural Security is not being used, the Construct Spectrum Administration subsystem manages users and their groups with its own set of security tables. The security data is always synchronized with any updates made to table data, whether the updates are made to data in the Construct Spectrum Administration subsystem or data in the Natural Security system.

Using Natural Security instead of Spectrum's built-in security allows you to use security definitions that are already in place. There is no need to duplicate this information in the Construct Spectrum Administration subsystem files. The subsystem's functionality is provided as an alternative for those users without access to a Natural Security system.

For more information about enabling Natural Security in the Construct Spectrum Administration subsystem, see **Using the Construct Spectrum Control Record**, page 89.

For information about defining users and groups when Natural Security is enabled, see **Defining Groups and Users**, page 95.

## Configuring Natural Security for Construct Spectrum

If you wish to use Natural Security to maintain users and groups, you have the option of configuring your Natural Security environment so that it can inform Construct Spectrum whenever changes are made to the Users and Groups tables in Natural Security. This ensures the integrity of the data in Construct Spectrum's security cache. For example, if a user is unlinked from a group in Natural Security, all records associated with that user in Construct Spectrum's security cache will be deleted.

- To configure the Natural Security environment, follow these steps:
  - 1 Copy the module NSCUSEX1 from the SYSSPEC library into SYSSEC library. If you are already using the NSCUSEX1 exit in Natural Security, you can copy the sections of code identified in the NSCUSEX1 module supplied with Construct Spectrum into your existing version of NSCUSEX1.
  - 2 Use Natural Security to add library SYSSPEC as a steplib to library SYSSEC.
  - 3 Catalog module NSCUSEX1 in library SYSSEC.
  - 4 Modify the NATPARM that you use to start your Natural Security sessions to include the LFILE definitions required by Spectrum (LFILE 135 and 136).

## Starting Services in Batch Mode under Natural Security

When you are starting Spectrum services as a batch job under Natural Security, refer to the guidelines supplied in the *Natural Security Manual for Mainframes*. For example, use `STACK=(LOGON[library] [user-id] [pswd])` in the NATPARM settings used to start the Natural session in the batch job.

The user-id that is used to perform the Natural Security logon for batch jobs starting Spectrum must be linked to library SYSSPEC. No other special privileges need be granted to the user.

- To allow Construct Spectrum to communicate with Natural Security:
  - 1 Access the Natural Security main menu.
  - 2 Invoke the Administration Services main menu.
  - 3 Invoke the General Options menu.
  - 4 Set the “Free access to functions via interface subprograms” property to TRUE.

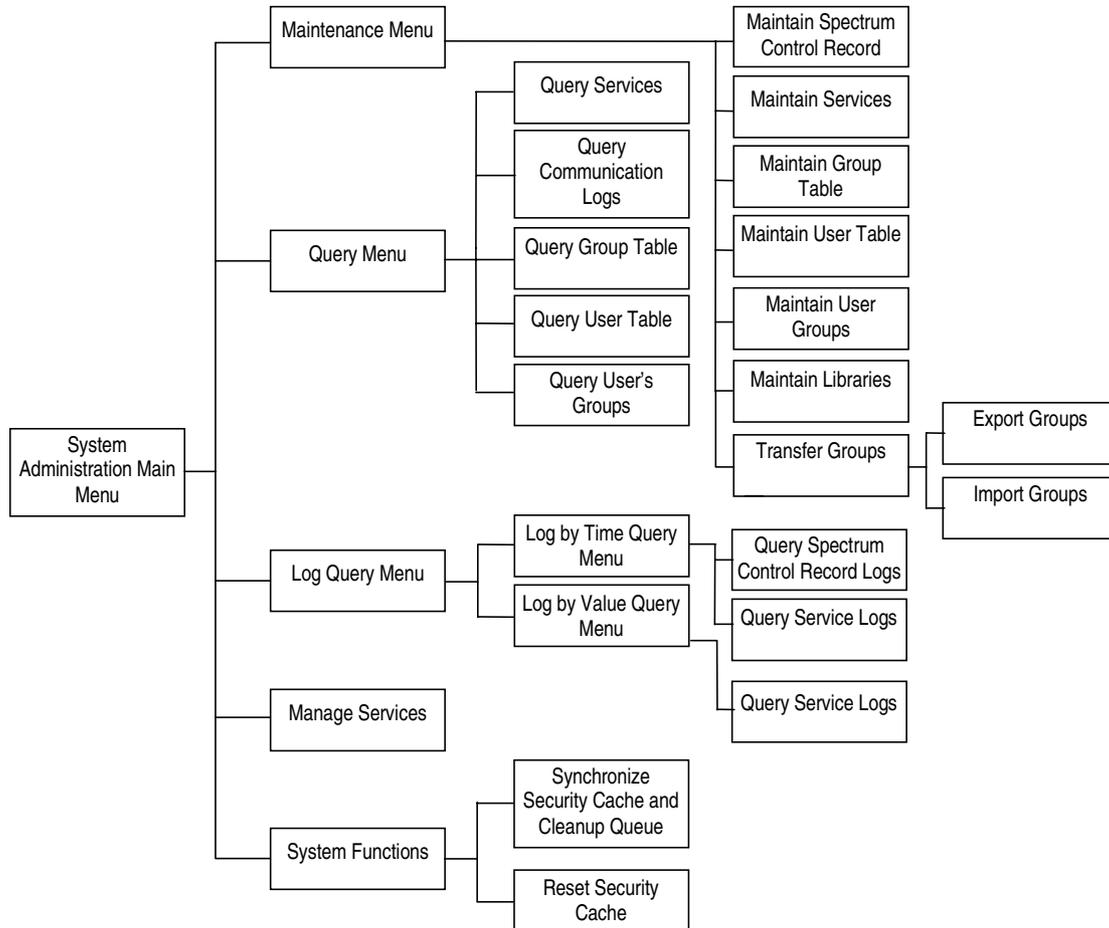
## Natural Security Interface to Restricted Libraries

Construct Spectrum uses the published Natural Security interface, NSC---L, to determine whether a user may use a library. This interface does not provide a means to specify a DBID and FNR along with the library name. This means that, if a user is linked to a library through Natural Security on a given DBID or FNR, Construct Spectrum cannot verify that the library it is accessing on behalf of the user is on the same DBID or FNR that is specified under Natural Security.

When you define steplib chains in the Construct Spectrum Administration subsystem, be sure to specify only libraries that are protected by Natural Security. That is, when you specify a steplib with a DBID or FNR in Construct Spectrum, the library should be defined in Natural Security with the same DBID or FNR. Note that you cannot use the same name for libraries on different DBIDs or FNRs.

## System Administration Menus and Natural Security

When Natural Security is enabled in the Construct Spectrum Administration subsystem, the options on the System Administration menus differ slightly. The following diagram shows the menu structure:



System Administration Menu Structure when Natural Security is Enabled

If you choose to use Natural Security as your user and group management mechanism, the following changes appear in the System Administration menus:

### **System Administration Maintenance Menu**

- The Maintain Users function invokes the Natural Security user maintenance panel.
- The Maintain Groups function invokes the Natural Security group maintenance panel.
- The Maintain User Groups option is available as a way to associate users and groups.
- The Maintain Libraries option is available as a way to define links between users and libraries.

### **System Administration Query Menu**

- The Query Users function invokes a Natural Security user information window.
- The Query Groups function invokes a Natural Security group information window.
- The User's Groups query is available.
- Queries for User and Group log information (both by time and by value) are not available.

## Using EntireX Security with Construct Spectrum

If you use EntireX Security to manage a security system (such as RACF, CA-TOP SECRET, or ACF2), you can use your established security system to authenticate users accessing Construct Spectrum applications. Construct Spectrum supplies a module called SPSSAF in the SYSSPEC library that CallNats a routine to perform authorization checks based on object, domain, method, and user ID.

Configuring EntireX Security for Construct Spectrum applications involves the following steps:

- 1 Add a new resource class or type called “SPECTRUM”. If you call it something other than “SPECTRUM”, you must change the reference to the class or type in SPSSAF.
- 2 Create resource profiles for each domain-object-method combination used in the application, using the format `domain.object.method`.

*Note: Make sure that the number of characters specified in resource profiles do not exceed 32. Do not use special characters in your resources.*

- 3 Add the resource profiles to the resource class or type.
- 4 Set up your Spectrum dispatch service to use the SAF security mode. For more information, see **Defining Construct Spectrum Services**, page 51.

For more information about defining resources to your security system, see the EntireX Security documentation.

*Note: If you modify the resource profile once the Construct Spectrum application is in use, be sure to reset the Spectrum security cache to clear previous permissions from memory. For more information, see **Resetting the Security Cache**, page 142.*



---

## DEFINING AND MANAGING CONSTRUCT SPECTRUM SERVICES

Construct Spectrum services automate the interaction between application components and Entire Broker, freeing application developers from the task of coding calls. Spectrum services also control such application-independent facilities as security, compression, encryption, data translation, and error recovery.

Construct Spectrum provides predefined dispatch, security, and attach services that you can modify to suit your needs. You can also create your own Spectrum services and choose how services are initiated (for example, in individual batch jobs or as subtasks of an attach service).

Once you have modified or created Spectrum services for your system, you can test, initiate, shutdown, and “ping” them (determine if they are running). You can also view query information about Spectrum services.

The following topics are covered:

- **Types of Spectrum Services**, page 48
- **Defining Construct Spectrum Services**, page 51
- **Managing Construct Spectrum Services**, page 72
- **Queries and Logs for Construct Spectrum Services**, page 84

## Types of Spectrum Services

There are three types of Spectrum services, as explained in the following table:

| <b>Service Type</b> | <b>Description</b>  |
|---------------------|---|
| Attach              | <p>An attach server starts other services. When defining dispatch, security, or user-defined services, you can associate them with an attach service. If a server is intended to execute as a subtask, the attach server must be executing before the services associated with it can be started. Attach services can be used to automatically start other services on demand.</p> <p>For more information, see <b>Attach Servers</b>, page 57.</p>           |
| Server              | <p>A server is a program that registers a service with Entire Broker and enters a receive loop waiting for clients to request its services. Designating a program as a server causes the common start facility (SPSSTART) to automatically LOGON and register with Entire Broker. Servers can also act as clients.</p> <p>Servers implement two kinds of services: main and command. For more information, see <b>Main and Command Services</b>, page 49.</p> |
| Client              | <p>A client is strictly a user of services. Client services do not register services with Entire Broker but request the use of services by sending a message to a server. When clients are started using the common start facility (SPSSTART), no implicit commands are executed. Client services are normally used only to define a Broker ID, userid, and password required to LOGON to Entire Broker.</p>  |

---

## Main and Command Services

All Spectrum servers implement two different services: a main service and a command service. The main service is defined on the Maintain Services panel, where the service field always contains the value MAIN. The command service uses the same server class and server name, but it uses the service CMD. For example:

|                 | <b>Server Class</b> | <b>Server Name</b> | <b>Service</b> |
|-----------------|---------------------|--------------------|----------------|
| Main service    | SPECTRUM            | DISPATCH           | MAIN           |
| Command service | SPECTRUM            | DISPATCH           | CMD            |

The main service implements commands that are specific to the server. The command service implements commands that are independent of servers; that is, commands that are generic and applicable to all servers, such as ping and shutdown. Having separate services allows you to automate the server startup process in response to some services but not others. For example, a dispatch server should be automatically started when a dispatch request is received, but commands such as ping and shutdown should not cause a new server to be launched.

The following table explains the command services:

| <b>Command</b> | <b>Description</b>  |
|----------------|---|
| PING           | Determines whether a service is available                                     |
| ENVIRONMENT    | Retrieves information about the environment in which the service is executing |
| TRACE          | Sets trace options to help debug a service                                    |
| CALLNAT        | Executes a user-supplied subprogram   |
| SHUTDOWN       | Ends a service  |
| ERROR          | Triggers a runtime error in the server  |

| <b>Command</b> | <b>Description (continued)</b> |
|----------------|--------------------------------|
| STEPLIBS       | Shows current steplib settings |
| SYSPROF        | Shows current LFILE settings   |

*Note:* When sending command services programmatically, use the syntax `CMD command`, where `command` is one of the commands listed in the previous table.

## Defining Construct Spectrum Services

Using the two Maintain Services panels in the Spectrum Administration subsystem, you can create three types of services:

- attach
- server
- client

This section explains how to access and use the Maintain Services panels. It also explains how to add information to the Attribute file and create a Natural profile to initialize a service at runtime.

### Accessing the Maintain Services Panels

- To access the Maintain Services panels:
- 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “MM” in the Function field of the System Administration Main Menu.  
The System Administration Maintenance Menu is displayed.
  - 3 Enter “SE” in the Function field of the System Administration Maintenance Menu.  
The first Maintain Services panel is displayed.

## Overview of the Maintain Services Panels

This section shows the panels and explains the types of settings available on them. The sections that follow explain how to use the settings.

```

SPBSMD      ***** Construct Spectrum Administration Subsystem *****      SPBSMD11
Aug 24      - Maintain Services -                                          1 more >

  Action (A,B,C,D,M,N,P)  _ Name: _____
                          Desc: _____

+- Entire Broker Service Settings -----+
| Broker ID.....: _____|
| Server class.....: _____|
| Server name.....: _____|
| Service.....: _____|
| User ID.....: _____|
| Password.....: _____|
| Attach Service name.....: _____ * Attach...: _|
+-----+
+- Service Start Parameters -----+
|>|
|>|
|>|
| Service start routine....: _____ *      Source      Object|
+-----+
Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
confm help  retrn quit      flip pref      left right main

```

### Maintain Services Panel 1

You can use this panel to identify the service, specify Entire Broker service settings, and define Service Start parameters. Not all of these settings are appropriate to all types of services, as explained below:

- Entire Broker service settings configure the Spectrum service to register itself by a specific name with a specific Broker. You must set settings for services of type A (Attach) and S (Server). Note that services that are linked to an attach service must specify the same Broker ID as the attach service. Therefore, when the Broker ID of an attach service is modified, the Broker ID of all related services is also automatically updated.
- Most services are assigned a start routine to initiate the service in an individual batch job, or through an attach service. Two start routines are supplied with Construct Spectrum. See **Using Service Start Routines**, page 60 for information about using and creating start routines for the different types of services.

Use the second Maintain Services panel to define the settings that internally configure the Spectrum service. These internal configurations include the service timeout value, default language, and program parameters. You must define some or all of these settings depending on the Spectrum service type.

```

SPBSMD      ***** Construct Spectrum Administration Subsystem *****      SPBSMD21
< 1 more   - Maintain Services -                                           4:58 PM

  Action (A,B,C,D,M,N,P)  _ Name: _____
                        Desc: _____
+- Spectrum Service Settings -----+
| Service type.....: _ (Server, Attach server, Client)
|
| Default language.....: __
|
| Service timeout.....: _____
|
| Unique ID.....:
+-----+
+- Program Parameters -----+
|>
|>
|>
| Natural program name....: _____      Source      Object
+-----+
Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
confm help  retrn quit          flip pref          left right main
Scrolling performed.
    
```

Maintain Services Panel 2

## Identifying the Spectrum Service

Identify and describe the Spectrum service on the first Maintain Services panel, using the following fields:

| Field       | Description  |
|-------------|--|
| Name        | In the Name field, supply a unique identifier for the Spectrum service.  |
| Description | The description can be any string that assists you in identifying the service. Be sure to make the description distinctive, since it is used in other panels to help identify the service. |

## Defining Entire Broker Service Settings

The Entire Broker Service Settings are located on the first Maintain Services panel:

```

+- Entire Broker Service Settings -----+
| Broker ID.....: _____|
| Server Class.....: _____|
| Server name.....: _____|
| Service.....: _____|
| User ID.....: _____|
| Password.....: _____|
| Attach Service name.....: _____ * Attach... _|
+-----+

```

Broker Service Settings

By filling in the Entire Broker service settings fields, you provide the parameters the Spectrum service needs to register with Entire Broker. On startup, a service uses the parameters to tell Entire Broker that it is running and is ready to handle requests. Later, you will add some of this information to the Entire Broker Attribute file. (For more information, see **Changing the Attribute File**, page 70.)

| <b>Field</b> | <b>Description</b>  |
|--------------|---|
| Broker ID    | <p>Each installation of Entire Broker is assigned a three-digit number. This number uniquely identifies the Entire Broker to your network. If you do not know the Broker ID, see your organization's network administrator.</p> <p>Specify a Broker ID that contains the three-digit number assigned to the Entire Broker being used. For example, if you are using an Entire Broker designated as number 45, the Broker ID specified must contain "045" somewhere in the ID. The Broker ID could be BKR045, BID045, or B045NODE. It is important to ensure that the Broker ID contains only one such numeric component.</p> <p>If a server (type S) specifies an attach server, there is no need to supply a Broker ID because this is automatically assigned to match the Broker ID of the attach server.</p> |
| Server class | <p>The server class value is normally used to identify the environment where the Spectrum servers are running. For example, SPECTRUM-PROD might be used to identify Production services, while SPECTRUM-TEST could designate services running in a test environment.</p>  |

| Field                         | Description (continued)   |
|-------------------------------|---|
| Server name                   | <p>The server name value is normally used to identify a specific program that offers services. Rather than using a physical program name, use the server name to define a logical name for the server. This allows you to change the physical program name without affecting any of the client programs that use the service.</p> <p>For more information, see <b>Server Names Supplied with Construct Spectrum</b>, page 57.</p> |
| Service                       | <p>The service name cannot be modified; it is always assigned the value "MAIN". For more information, see <b>Main and Command Services</b>, page 49.</p> <p>Note that the server class/server name/MAIN service and the server class/server name/CMD service must be defined in the Entire Broker attribute file for the specified Broker. For more information, see <b>Changing the Attribute File</b>, page 70.</p>             |
| User ID and Password          | <p>The user ID and password values correspond to the values that the server will use to assign the corresponding fields in the Entire Broker control block when making calls using the Entire Broker ACI (Advanced Communication Interface).</p>  |
| Attach service name<br>Attach | <p>For services of type "S" that are to be initiated automatically on demand, specify the attach service that will be used to launch the service and mark the Attach field.</p> <p>If the service is to be initiated manually, supply an attach service name but do not mark the Attach field.</p> <p>For more information about attach servers, see <b>Attach Servers</b>, page 57.</p>  |

## Server Names Supplied with Construct Spectrum

The following table shows the server names that are supplied with Construct Spectrum, along with the Natural program name that implements the service:

| Server Name | Natural Program | Description               |
|-------------|-----------------|---------------------------|
| ATTACH      | SPSATT          | Attach service            |
| DISPATCH    | SPSDISP         | Dispatch CALLNAT requests |
| SECURITY    | SPSSEC          | security service          |
| TIMESTAMP   | SPSTIMS         | Time stamp demo service   |

### Attach Servers

Entire Broker supports an automated server startup mechanism by means of an attach server. An attach server typically registers its own service with Entire Broker and then registers other services that it is capable of starting using an attach option. When a request is made for an attached service and no such service is currently available (waiting in a receive state), Entire Broker notifies the attach server that a request is pending. Normally, that attach server launches a new server to satisfy the request.

### Linking Services to an Attach Server

An attach service must be defined in the Spectrum Maintain Services panels using Service type “A” (attach). All services that the attach server is to automatically launch must specify the name of the attach service and must have the Attach field marked. Marking the Attach field tells the attach server to register the current main service with Entire Broker using the Attach option. Specify the attach server name without marking the Attach field for servers that are only to be launched manually. The combination of Server class, Server name, and Service must be unique for all services that are linked to a specific attach server with the Attach field marked.

An attach server can launch services either as separate batch jobs or as subtasks of the attach server. For more information, see **Using Service Start Routines**, page 60.

Note that the attach server only registers the main service of servers it is capable of launching. Therefore, commands such as Ping and Shutdown (which go to the command service) will not cause a server to be launched.

### **Activation Policies**

Every time an attach server is called by Entire Broker to satisfy an attach request, the attach server launches one instance of the requested service. This server will continue to run indefinitely as long as it is satisfying service requests. If the server stays idle for longer than its timeout period (specified on the second Maintain Services panel), it will shut itself down.

### **Runtime Error Handling**

If a server receives a runtime error, its restart policy depends on whether it is linked to an attach server. If the server is not linked to an attach server, it attempts to recover from the error. Otherwise, the server shuts down as a result of the error.

### **Common Server Startup Interface**

The main interface for starting Spectrum services is the START command. This section explains the different parameters you can use with the START command, depending on the type of service and the method of initiating it.

For Services of type S or A, the START command takes care of logging on to Entire Broker and registering the required Entire Broker Services. It then invokes the Natural Program entered on panel 2.

### Using the START Command to Initiate Services as Subtasks

When you are using an attach service to initiate a server or client service, the START command does not require any parameters. The system automatically uses the name of the service being defined. For example, the following parameters can be used to start the current service as a subtask of an attach server.

```
+-- Service Start Parameters -----+
|> NATPARAM='FNAT=(1,2),PROFILE=SYSSPEC
|> STACK=(LOGON SYSSPEC;START) '
|>
| Service start routine...: SUBTASKB *      Source SYSSPEC  Object SYSSPEC
+-----+
```

#### Example of Using the START Command

### Using the START Command to Initiate Services Online or in Batch

If you wish to start a service online (from the NEXT prompt) or in an individual batch job, the START command must have at least one parameter: the name of the service to use for configuration on startup. For example,

```
START DISPATCHER
```

where DISPATCHER is the name of the service definition to be used for configuration.

*Note: The START command is just a front-end to the program SPSSTART. When executing servers from the outside of the SYSSPEC library, use SPSSTART rather than SMRT to ensure access to the proper environment.*

## Defining Service Start Parameters and Service Start Routines

The Service Start Parameters and Service start routine fields are located at the bottom of the first Maintain Services panel. The service start parameters differ depending on the service start routine you use: BATCHJOB or SUBTASKB. See **Using Service Start Routines**, page 60 for more information.

Although the service start parameters are shown on three separate lines, internally these lines are treated as one long string value. Therefore, it is possible to define parameters that span multiple lines. To do so, simply continue the parameter on the next line. For example,

```
+-- Service Start Parameters -----+
|> JCL=BATCHTXT, JOB-PREFIX=AS, NATPARM=' IM=D, PROFILE=SYSSPEC, STACK=
|> (LOGON SYSSPEC, MYUSERID, MYPWD) '
|>
| Service start routine...: BATCHJOB *           Source SYSSPEC  Object SYSSPEC
+-----+
```

### Example Service Start Parameters

We recommend that you use commas to delimit parameters, although you can also use blank characters. If a parameter ends on the last character of a line, and it is not to be concatenated with the parameter beginning on the next line, include a comma (or blank character) at the beginning of the next line.

## Using Service Start Routines

The Service start routine field is located on the first Maintain Services panel. All servers must be assigned a start routine. The start routine defines how the Natural session that will host the server is to be initiated. Two start routines are supplied with Construct Spectrum in source form. One or both of these can be used in most environments, or you can use these as examples for writing your own routine. The supplied routines are:

| Start routine | Description   |
|---------------|---|
| BATCHJOB      | Calls NATRJE to launch a batch job to host the service.   |
| SUBTASKB      | Calls CMTASK to start a Natural subtask. This routine can only be used in Natural environments that support subtasks. In such environments, BATCHJOB is typically used to start attach servers and SUBTASKB is used to start the other servers. |

## BATCHJOB Service Start Routine

This service start routine is used to launch a separate Natural batch job to host the server. The service start parameters must specify the name of the Natural source member that contains the JCL to be submitted. This source member is identified by the JCL keyword. All attach servers are started using the BATCHJOB start routine.

To reduce the number of JCL members that are required, the JCL can contain many different place holders that can be substituted during the job submission process.

The following table shows the service start keywords that are valid for the BATCHJOB service start routine.

| Keyword    | Description  |
|------------|--|
| JCL        | <p>The name of the Natural source member that contains the JCL to be submitted. The member must exist within the library where the attach server that launches the service is running (usually the SYSSPEC library).</p> <p>The following JCL members are provided as templates in the library SYSSPEC. You can copy and modify these members to create a customized BATCHTXT member.</p> <ul style="list-style-type: none"> <li>• BATCHJCL (MVS example)</li> <li>• BATCHDCL (DOS VSE/ESA example)</li> <li>• BATCHBCL (BS/2000 example)</li> </ul> <p>If you place &amp;keyword place holders in the specified JCL text member, you can use the following keywords.</p> <p>For more information, see <i>Construct Spectrum and SDK Mainframe Installation</i>.</p> |
| JOB-PREFIX | <p>If the JOBNAME is to be uniquely generated, the two-character JOB-PREFIX can be specified on each service to help associate the job name with the Spectrum service. This value defaults to 'SP'.</p>  |

| Keyword     | Description (continued)   |
|-------------|---|
| JOB-NAME    | <p>If you want to explicitly name the job, use the JOB-NAME keyword rather than the JOB-PREFIX keyword. Note that some operating systems do not allow multiple jobs with the same name to be run concurrently. Therefore, using the JOB-NAME keyword rather than JOB-PREFIX may cause a conflict.</p> <p>In the JCL template, the keyword &amp;JOB-PREFIX&amp;JOB-NR is used to represent the JOB-NAME.</p> |
| TRANSACTION | <p>The name of the Natural nucleus under which the service will run. This nucleus must have had Spectrum's load modules linked to it. The default transaction is NAT232RE for MVS and NAT232BA for other environments. It is also possible to hard-code the desired transaction in the JCL member.</p>  |
| NATPARM     | <p>The Natural parameters to be used when initiating the service. The default NATPARM value is 'IM=D, PROFILE=SYSSPEC'.</p>   |
| LIBRARY     | <p>The name of the library to which you wish to logon. This value defaults to SYSSPEC.</p>  |
| USER-ID     | <p>The Natural user-id to be used to log on to Natural. This keyword only applies to server running under Natural Security.</p> <p>If the USER-ID matches the User ID you specify in the service settings, you can use *USER-ID.</p>  |

| Keyword  | Description (continued)   |
|----------|---|
| PASSWORD | <p>The Natural Security password associated with the USER-ID.</p> <p>If this password matches the one you specified in the service settings, you can use *PASSWORD, thus preventing the password from being displayed on the current screen.</p>  |
| PROGPARM | <p>If the server program you are invoking accepts additional parameter values, these can be substituted for the PROGPARM keyword. These parameters can even be resubstituted with actual parameter values passed into SPCATT, which is used to launch a remote service from a client.</p> |

### Example BATCHJOB Routine

In most cases, the default values defined in the BATCHJOB program or hard-coded in the JCL can be used. Therefore, the parameters to start up a batch job typically appear as follows:

```
JCL=BATCHTXT, JOB-PREFIX=XX
```

If the server runs under Natural Security, you may also wish to include a user-ID and password. This example uses the same user-ID and password that are specified in the Entire Broker service settings:

```
Service.....: MAIN
User ID.....: TECH1
Password.....:
```

This example supplies a full JOB-NAME, rather than just the prefix. It also overrides the NATPARMS. Notice that statement must be embedded in quotes because it contains special characters.

```
JCL=MYJCL, JOB-NAME=TESTJOB, USER-ID=*USER-ID, PASSWORD=*PASSWORD,
NATPARM=' IM=D, FNAT= (1, 2) , PROFILE=QA'
```

## SUBTASKB Service Start Routine

The SUBTASKB service start routine accesses, by means of Entire Broker, an attach service running in batch. The request to start the specific task is transmitted to the attach service. The attach service uses the data to start the requested service as an attached task inside the batch region.

Use this service start routine if you want the service to be executed as a subtask of an attach server. In this case, you must also specify an attach service name, and the attach service must be executing in order to launch the current service.

For operating systems that support Natural's subtasking feature, this is the preferred approach because fewer batch jobs need to be initiated.

When using SUBTASKB, you must specify the parameters required to start the subtask. The STACK parameter should include a LOGON to the SYSSPEC library and the required stack program. If you simply want to launch the current server, the START keyword should be used.

The STACK parameter differs depending on whether the servers run under Natural Security and depending on the AUTO parameter specified in your NATPARMS.

The SUBTASKB service start routine supports the following:

| <b>Keyword</b> | <b>Description</b>  |
|----------------|---|
| TRANSACTION    | The name of the Natural nucleus under which the service will run. This defaults to the same transaction as used by the attach server that launches the subtask. |
| NATPARAM       | The Natural parameters to be used when initiating the service. These parameters can be any of the values allowed when initiating a Natural session.             |

### Example SUBTASKB Routine

The NATPARAM parameter is usually the only required parameter. The format of this parameter varies depending on whether or not you are executing under Natural Security.

### ***SUBTASKB Parameters Without Natural Security***

```
NATPARM= ' FNAT= (1020,1000) , PROFILE=SYSSPEC,  
STACK= (LOGON SYSSPEC;START) '
```

Note that the NATPARM parameter must be in quotes because it contains special characters. In this example, the START keyword has been used to invoke the common server startup facility. Once Natural is started, the Natural Program Name defined on the second Maintain Services panel will be invoked.

### ***SUBTASKB Parameters Under Natural Security***

Under Natural Security, the Natural startup parameters depend on the value of the AUTO NATPARM. This example uses AUTO=OFF so that the user-ID and password must be supplied. You can either enter a valid Natural user-ID and password or used the special keywords \*USER-ID and \*PASSWORD to use the same values that were supplied in the Broker Service Settings portion of the first Maintain Services panel.

```
NATPARM= ' FNAT= (1,2) , PROFILE=SYSSPEC, AUTO=OFF,  
STACK= (LOGON SYSSPEC, *USER-ID, *PASSWORD;START) '
```

### **Writing to Print and Workfiles**

It is possible for programs executed by a dispatch service running in batch to write information to a print file or workfile. When doing so, be sure that the batch job which initiates the service (this would be the attach job if you are using an attach service) defines the necessary output devices. For example, CMPRTnn and CMWKFmm.

### **Generating Trace Output**

If you are experiencing problems initiating or running Spectrum services, you can trace the flow of the Construct Spectrum software to help uncover the problem or to provide additional information to SAGA SOFTWARE support. To generate trace output, stack the command TRACE prior to initiating the desired service.

The following parameters show an example of how tracing can be enabled.

```

SPBSMD      ***** Construct Spectrum Administration Subsystem *****      SPBSMD11
Sep 16      - Maintain Services -                                          1 more >

  Action (A,B,C,D,M,N,P)  _ Name: DISPATCH_____
                        Desc: Spectrum Dispatch Server_____
+- Entire Broker Service Settings -----+
| Broker ID.....: BKR123_____
| Server class.....: SPECTRUM_____
| Server name.....: DISPATCH_____
| Service.....: MAIN
| User ID.....: SPSDISP_
| Password.....:
| Attach Service name.....: ATTACH_____ * Attach...: X
+-----+
+- Service Start Parameters -----+
|> NATPARAM='FNAT=(1,2),PROFILE=SYSSPEC,
|> STACK=(LOGON SYSSPEC;TRACE;START)'
|>
| Service start routine....: SUBTASKB *          Source SYSSPEC  Object SYSSPEC
+-----+
Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
confm help  retrn quit          flip  pref          left  right main

```

### Example of Using the TRACE Command

A number from one to five can also be passed to the TRACE command to cause the trace output to be written to CMPRT01 through CMPRT05, respectively. Or, “W1” through “W5” could be passed to the TRACE command to cause the trace output to be written to CMWKF01 through CMWKF05, respectively.

*Note: It is possible to write additional service start routines that are applicable to other environments. Use the supplied programs BATCHJOB and SUBTASKB as templates.*

## Defining Spectrum Service Settings and Program Parameters

The purpose of the Spectrum service settings is to internally configure the service. The Spectrum service settings and program parameters are located on the second Maintain Services panel:

```

SPBSMD      ***** Construct Spectrum Administration Subsystem *****      SPBSMD21
< 1 more          - Maintain Services -          4:58 PM

  Action (A,B,C,D,M,N,P)  _ Name: _____
                        Desc: _____
+- Spectrum Service Settings -----+
| Service type.....: _ (Server, Attach server, Client)
|
| Default language.....: _
|
| Service timeout.....: _____
|
| Unique ID.....:
+-----+
+- Program Parameters -----+
|>
|>
|>
| Natural program name....: _____      Source      Object
+-----+
Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
confm help  retrn quit      flip  pref      left  right main
Scrolling performed.
    
```

Maintain Services Panel 2

These fields have the following functions:

| <b>Field</b>     | <b>Description</b>   |
|------------------|--|
| Service Type     | Specify "A" when defining an attach server, "S" when defining a normal server, and "C" when defining a client service. The START program automatically performs a Broker LOGON and Register for services of type A or S.   |
| Default Language | <p>This is the value that will be assigned to the Natural system variable *LANGUAGE on server startup.</p> <p>The service uses this value to determine which language to use when returning messages to the client. The client can override this value when making a request, but the service always returns to this default value after servicing a client.</p>   |
| Service Timeout  | <p>This value specifies the amount of time that a server will continue to execute after the last time it serviced a client request.</p> <p>Normally, attach servers should never be allowed to time out because they are required to launch all other servers. Therefore, a very high timeout value, such as 9999H, should be applied to attach servers.</p> <p>The only problem with allowing other servers to time out is that users may experience a slight delay whenever a new server must be started to satisfy a request. Nevertheless, you do not want to have very large timeout values because this may result in many idle servers consuming system resources. For most servers, 30 to 60 minutes is recommended.</p> <p>The service timeout value is specified as an entirely numeric value or as a number suffixed with an interval. If there is no interval suffix, the number is assumed to specify seconds. Valid interval suffixes are:</p> <ul style="list-style-type: none"> <li>• S seconds (for example, 3600S)</li> <li>• M minutes (for example, 60M)</li> <li>• H hours (for example, 3H)</li> </ul> |

| Field              | Description (continued)  |
|--------------------|--|
| Unique ID          | Each service requires a 16-character unique ID. This derived value is used as the prefix of the TOKEN field in the BROKER control block. This allows Spectrum to associate a running service with a Spectrum service record.   |
| Program Parameters | <p>Some servers require additional parameters to configure themselves. These parameters are placed in the Program Parameters area.</p> <p>The following program parameters are required by the supplied servers:</p>   |
| Dispatch Service   | <p><code>security-SERVER=service-name</code></p> <p>This is the name of the service that the dispatch service communicates with to retrieve security information.</p> <p><code>security-MODE=value</code></p> <p>This parameter is used to disable/enable security checking by specifying the security mode. When security is disabled, the security server is only used to look up program names and library information. When enabled, the security service uses the security mode specified to initiate the security check. This parameter can have the following values:</p> <p>NONE - No security, program names and libraries are checked.</p> <p>SPECTRUM- Spectrum authenticates users and checks authorizations.</p> <p>NSC- Natural Security authenticates users and checks authorizations.</p> <p>SAF- SAF authenticates users and checks authorizations.</p> <p>By default <code>security-MODE=NONE</code>. If you change the value of the <code>security-MODE</code>, ensure you clear the security Cache record.</p> |

| Field                | Description (continued)   |
|----------------------|---|
| Attach Client        | CONVERSATION-FACTORY=service<br><br>This parameter identifies the service record that defined the Broker settings to be used to generate conversations when using the SQU* API. |
| Natural program name | When the START command is used, this field identifies the program that will be invoked by the common start facility to implement the service.                                   |

## Changing the Attribute File

The values specified in the Server class, Server name, and Service fields on the first Maintain Services panel must also be specified in the Entire Broker Attribute file for each Spectrum service. These attributes are necessary for Entire Broker to identify Spectrum services and their characteristics.

In addition, you must add the Entire Broker service definitions required by the command services. These are the same as the main service, with the service name of CMD instead of MAIN.

*Note: Each Entire Broker has its own Attribute file, so be sure to add the service definition to the appropriate Attribute file.*

You do not have to change the Attribute file until just before you start the service. Before that point, someone with the proper access authority must update the Entire Broker Attribute file.

For more information about the Attribute file, see **The Attribute File** in the *Entire Broker Reference Manual*.

*Note: Spectrum requires that the Entire Broker attribute TRANSLATION not be specified for dispatch and security services. If you are using the Entire Broker V4.1.1, you must set the attribute DEFERRED=NO as the broker default.*

## Adding a Natural Profile

You can use Natural profiles to fully define a Natural environment. They are useful for reducing the amount of information you must supply when starting a Natural session. These profiles can be set up using the Natural SYSPARM utility. (For more information about the Natural SYSPARM utility, see the *Natural Utilities Manual*.)

With the Construct Spectrum Administration subsystem, profiles can be useful when you are defining Spectrum services. As part of defining a service, you must fill in the Service Start parameters. Use the NATPARM keyword to specify the system file used to store the Natural profile.

For an example of creating a Natural profile, see **Step 14 — Create a New Natural Profile**, page 43, *Construct Spectrum and SDK Mainframe Installation Manual*.

## Managing Construct Spectrum Services

The main purpose of the Manage Services panel is to initiate attach servers. Since all other services can be initiated automatically (on demand) by an attach server, it is usually not necessary to initiate other servers in advance of their use. When servers fail to start, the Manage Services panel can also be used to help diagnose the problem. Additionally, this panel is used to send inquiry commands to various servers to determine the environment in which they are running. Finally, the Manage Services panel can be used to shut down servers.

## Accessing the Manage Services Panel

- To access the Manage Services panel:
  - 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “MS” in the Function field of the System Administration Main Menu.  
The Manage Services panel is displayed:

```

SPCMS          ***** Construct Spectrum Administration Subsystem *****          SPCMS0
Sep 17          - Manage Services -          8:15 AM

Action Spectrum Service          Type          Attach Manager
-----
___  ATTACH          Attach
___  DISPATCH        Server  ATTACH
___  security        Server  ATTACH
___  TIMESTAMP       Server  ATTACH
___  TST-ATTACH      Attach
___  TST-DISPATCH   Server  TST-ATTACH
___  TST-security    Server  TST-ATTACH
___  TST-TIMESTAMP   Server  TST-ATTACH
                                     *** End of Data ***

Service ..:: _____ Service type ..: _
Replica ID ..: _____ Repeat command: _
Last command:
Respondent ..

Direct Command: _____
Debug      EndJob      EndService  EnVironmnt  Initiate    Ping        Refresh
SHutdown   STeplibs   SYsprof    (PF4=CLog) (PF5=flip)
    
```

Manage Services

## Using the Manage Services Panel

Use the Manage Services panel to send commands to servers and display the replies returned from the servers. This panel can also be used to initiate new servers.

By default, this panel displays services of type S (server) and A (attach). To view client services, enter “C” in the Service type field, or enter \* in the Service type field to view all service types. In the Action column, you can enter the capitalized let-

ter(s) of a command shown at the bottom of the panel. These commands can also be used in the Broker Driver program. For more information, see **Broker Driver Program**, page 154.

The fields at the bottom of the panel have the following functions:

| <b>Field</b> | <b>Description</b>   |
|--------------|--|
| Service      | Use to reposition the display to a specific service.   |
| Service type | Enter "*" to show all services, leave the field blank to show services of type A and S, or enter a specific service type to be displayed.  |
| Replica ID   | <p>Entire Broker allows many different servers to provide the same Broker services. Each unique server is known as a replica. Each replica is identified by a unique 32-character replica ID. For most requests, the client does not care which replica satisfies the request. Occasionally, you may wish to target a specific replica. For example, you may wish to run a server online to output debugging statements (such as Input, Print, etc.) Later, when you are finished testing, you will want to shut down that specific replica. This can be done by including the desired replica ID as an option on the shutdown command. For example, using the SHUTDOWN command for a service without specifying a replica ID shuts down any replica offering the service. Specifying a replica ID and using the SHUTDOWN command causes all other servers to ignore the command. You must continue to send the command until the desired respondent replies.</p> <p>All commands support a replica ID option and the Respondent field indicates which replica responded to the command. Information about a replica can be retrieved using the Environment command.</p> |

| Field          | Description (continued)   |
|----------------|---|
| Repeat command | <p>This field allows you to execute a command multiple times. For example, to initiate three dispatch servers, type “I” beside the dispatch service and enter “3” in the Repeat command field.</p> <p>Repeating a Ping command can help determine how many servers are running because you can analyze the replica IDs of the respondents.</p> <p><i>Note: Using the Entire Broker Control Center is a better means of determining the number of servers available.</i></p> |
| Last command   | This field shows the text that was sent to the server to execute the most recent command. This string could also be sent to the server programmatically or using the Broker Driver program.   |
| Respondent     | This field identifies the Spectrum service and replica ID of the server that responded to the most recent command. The respondent is followed by the response text. Note that commands are always sent to the Broker service associated with the desired Spectrum service. If multiple Spectrum services specify the same Broker service, the command may be directed to a completely different Spectrum service than the one chosen.                                       |

You can also access the Communications Logs (PF4) to view system messages. For more information, see **Accessing the Query Communications Logs Panel**, page 83.

## Overview of Commands

Some commands are sent directly to the main service (MAIN) of the associated server. Other commands are sent to the command service (CMD) of the server. Still others are sent to the attach server associated with the server.

The following table summarizes the commands and how they are processed:

| <b>Command</b> | <b>Applies to...</b>  | <b>Sent to...</b>   |
|----------------|---|---|
| Debug          | services of type S or C that are linked to an attach server | MAIN service of the attach server   |
| EndJob         | attach servers that launch subtasks                         | MAIN service of the attach server. The attach server, in turn, sends Shutdown requests to linked services   |
| EndService     | services of type S that are linked to an attach server      | MAIN service of the attach server. The attach service sends Shutdown requests to the specified service  |
| EnVironment    | all services of type S or A                                 | CMD service of the Spectrum service   |
| Initiate       | all service types   | if the service has an attach server, the command will be sent to the attach server; otherwise, the service start routines will be called directly to initiate the service |
| Ping           | all services of type S or A                                 | CMD service of the Spectrum service   |
| Refresh        | all services of type A                                      | MAIN service of the attach server   |
| SHutdown       | all services of type S or A                                 | CMD service of the Spectrum server  |
| STeplibs       | all services of type S or A                                 | CMD service of the Spectrum server  |
| SYsprof        | all services of type S or A                                 | CMD service of the Spectrum server  |

Note that the Attach Service column shows the name of the attach server that is capable of starting the Spectrum service. The services showing the intensified attach server names are those that will automatically start in response to an Entire Broker attach request.

The following sections explain the commands in detail.

## Debug Command

Use the Debug command to help diagnose problems in starting services. The Debug command only applies to services that name an attach server.

The Debug command returns information related to the Natural startup process. When you use the START command, the START program detects if it is in debug mode and writes out additional diagnostics. In debug mode, the server is not intended to start up so the server program is not FETCHED.

### How the Debug Command Works

When a server is launched in debug mode, the CMPRINT dataset, where Natural writes initialization errors, is mapped to Work File 7 (CMWKF07). After launching the server, the attach server pauses for three seconds and attempts to read the contents of Work File 7. The contents of the file is sent back to the client.

*Note: Occasionally, you may experience I/O errors when using the Debug command. Normally, you can simply repeat the command.*

## EndJob Command

The EndJob command only applies to attach servers. More specifically, it applies to attach servers that launch other services as subtasks. When the EndJob command is sent to the attach server, the attach server sends shutdown commands to each service that it has launched.

As part of the Shutdown command, the attach server includes the PARENT-RID option so that only subtasks of the attach server respect the shutdown request.

Normally, the `EndJob` command causes the batch job that hosts the attach server and its subtasks to end. However, there may still be subtasks that are busy processing requests and therefore unable to receive the shutdown notification. In this case, the batch job will continue to run until all subtasks time out or are shut down explicitly.

## EndService Command

The `EndService` command can only be used with services that have an attach server. In response to this command, the attach server continues to send shutdown requests to the specified service until no further services respond.

Note that, while one attach server broadcasts the shutdown command to all replicas of the service, these replicas could be running within other jobs or environments. Unlike the `EndJob` command, the `EndService` command does not use the `PARENT-RID` to restrict the services that carry out the shutdown request.

## Environment Command

This command returns information about when a server was started as well as many of the Natural system variable values assigned within the server. This information can be used to help diagnose problems with a server and to identify the environment in which the server is running.

If the `Environment` command returns a non-blank `PARENT-RID` value, this indicates that the service is executing as a subtask where the `PARENT-RID` value identifies the attach server replica that hosts the service.

## Initiate Command

Use the `Initiate` command to start servers. You can start as many servers as needed, to the maximum defined in the `GLOBAL DEFAULTS` section of the `Entire Broker Attribute` file. If you wish to start a server or client service that uses an attach server, make sure that the attach server is already running. Follow the procedure described below to initiate servers.

- To start a server:
- 1 Enter “P” in the Action field of the service.  
If the system responds that the Ping action was successful, there is already a service of that name available on the network.  
If a message indicates that the service is not registered or if you decide that an additional copy is needed, continue to the next step.
  - 2 Enter “I” in the Action field of the service to initiate a copy.  
If the server runs as a batch job, note the name of the batch job when it is displayed because you may need it later. It can take up to 15 seconds (depending on system load) for the service to start and register with Entire Broker.
  - 3 Do one of the following:
    - Use the Ping action to determine whether or not it started properly.
      - If the Ping action is successful, the service is now available.
      - If the Ping action fails, press PF4 (CLog) to check the Communications Logs. These logs contain messages that are not communicated back to the client, such as startup errors. Use the information found here, along with the information in *Construct Spectrum Messages*, to determine and rectify the problem.
    - If this is not the first copy of the Spectrum service, repeat the Ping command to determine the number of replicas running based on the replica ID. If the server fails to start and the communication log (PF4) does not reveal why, perform one of the following steps.
      - For attach servers or services that are not started by an attach server:  
If an attach server fails to start, it is probably a problem with the JCL that is submitted. Review the batch job output and correct the JCL within the specified JCL member.
      - For services started by an attach server:  
Use the Debug command to try to identify any problems with the startup commands sent to Natural.

You can also determine the number of copies of a service that are running using the Entire Broker Control Center.

## Ping Command

Use the Ping command to determine whether or not services are running and available somewhere on the network. When a service is pinged, one of the following messages is returned:

| Message   | Description   |
|---|---|
| ETB Error 215/148<br>(NET: Connection Error)          | Indicates that the Broker ID that is specified in the service definition is not currently running or responding.  |
| ETB Error 7/7<br>(Service not registered)             | Indicates that you have requested a valid service, but there are no servers running to satisfy the request. If this error is preceded by a message such as "Sending command to attach server: <i>name</i> " this indicates that the attach server associated with the current server is not registered. |
| ETB Error 20/216<br>(API: Invalid BROKER-ID)          | Indicates that the Broker ID specified for the service is not valid.  |
| PING of Service<br><i>Service name</i> was successful | Indicates that at least one copy of the service that was pinged is running and available.   |
| ETB Error 74/74<br>(Wait timeout occurred)            | Either all services are busy or a service has ended without properly deregistering.   |

## Refresh Command

The Refresh command can only be sent to an attach server.

When an attach server is started, it determines all services to be registered with the Attach option by reading the Spectrum service records. While the server is running, it is possible to modify the service records and alter the set of services that are linked to the attach server. These changes are not reflected in running attach servers until an explicit refresh command has been sent to all such servers.

## Shutdown Command

There are three ways to shut down servers. The method you choose depends on whether you want to shut down a single replica, all replicas, or all servers within a single job.

### Shutting Down a Single Replica

To end a single server replica, use the SHUTDOWN command with the service to be ended. If you wish to shut down a specific instance of a server, also specify the replica ID.

### Shutting Down All Replicas of a Service

Multiple replicas of a service can be ended by sending the End Service (ES) request to the attach server that is used to start the service.

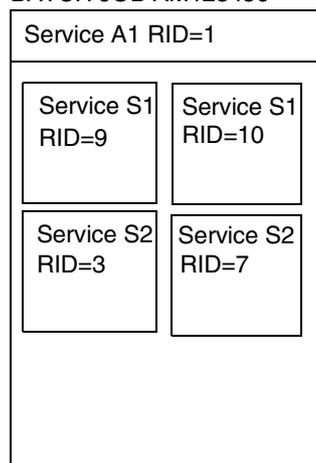
### Shutting Down All Servers Within a Batch Job

If your servers execute as subtasks of an attach server, you may wish to end all servers that are running within a certain batch job. To do this, enter the End Job (EJ) command in the Action field for the attach server to be shutdown.

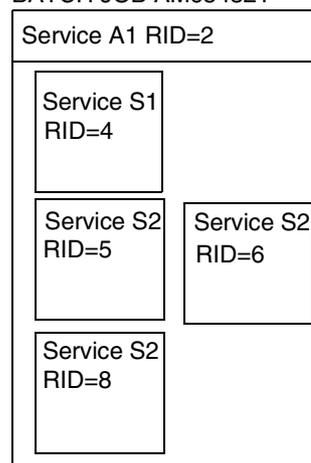
### Example of Shutting Down Servers in a Batch Job

In addition to using the Manage Services panel, you can also send commands directly to servers using the Entire Broker Driver program. This section describes commands that can be sent to servers to implement a variety of shutdown options.

BATCH JOB AM123456



BATCH JOB AM654321



- To end any replica of Service S2, you would send the command `CMD SHUTDOWN` to the service.
- To end Service S2 Replica 5, you would send the command `CMD SHUTDOWN, RID=5` to the service to be shut down until you receive a “successful” response.
- To end all replicas of Service S2, you would send the command `PROCESS SHUTDOWN SERVICE S2` to either attach service.
- To end all services beginning with S2, you would send the command `PROCESS SHUTDOWN SERVICE S2*` to attach service A1. Note that this command will shut down Service S2 regardless of which attach server S2 is running under.
- To end all services beginning with S2 and running batch job AM654321, you would send the command `PROCESS SHUTDOWN SUBTASK S2*, RID=2` to the attach service. Continue to send the command until it is received by Replica 2.
- To end batch job AM123456, you would send the command `PROCESS SHUTDOWN SUBTASK *, RID=1` to the attach service A1. Enter the command until it is satisfied by the desired replica.

## **STeplibs Command**

This command returns information about the Natural step libraries used by the server.

## **SYsprof Command**

This command returns information about the system files being used by a server.

## **Standard Response Protocol**

Most servers acknowledge or respond to standard protocol. This protocol is identified by the signature MSG111 at the beginning of the message.

## **Accessing the Query Communications Logs Panel**

You can access the Query Communications Logs panel directly from the Manage Services panel by pressing PF4 (CLog). Because each Initiate and SHUTDOWN action generates an entry in the Communications Logs, this PF-key gives you quick access to the messages.

The PF4 (MSrvc) key on the Query Communications Logs panel allows you to return to the Manage Services panel quickly.

## Queries and Logs for Construct Spectrum Services

In the Construct Spectrum Administration subsystem, you can view several queries and logs containing information about Construct Spectrum services. The following section explains the type of information each query and log shows and how to access the panels.

For information about accessing queries and logs for other Construct Spectrum Administration components, see the chapters that explain those components.

## Accessing Query Panels

- To access the Query panels:
- 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
- 2 Enter “QM” in the Function field of the System Administration Main Menu.  
The System Administration Query Menu is displayed:

```

BSSQMENU   ***** Construct Spectrum Administration Subsystem *****   CDLAYMN1
Aug 25     - System Administration Query Menu -                               02:13 PM

          Functions
          -----
          SE   Query Services

          CL   Query Communication Logs

          GR   Query Group Table
          US   Query User Table

          ?   Help
          .   Terminate
          -----
Function ..... _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit          flip                               main
    
```

System Administration Query Menu

The options on this menu that are relevant to Spectrum services invoke query panels that contain the following types of information:

| <b>Function</b>           | <b>Description</b>   |
|---------------------------|--|
| Query Services            | Lists existing Construct Spectrum services and their characteristics as defined on the Maintain Services panels. |
| Query Communications Logs | Lists messages that occurred during the operation of Construct Spectrum services.                                |

## Accessing Query Log Panels

- To access the Query Log panels:
- 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “LM” in the Function field of the System Administration Main Menu.  
The System Administration Log Query Menu is displayed:

```

BSSLMENU          Construct Spectrum Administration Subsystem      CDLAYMN1
Apr 15             System Administration Log Query Menu           04:47 PM

                  Functions
                  -----
                  LT   Log by Time Query Menu
                  LV   Log by Value Query Menu

                  ?   Help
                  .   Terminate
                  -----
Function .....  __

Command .....  _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retrn quit          flip                               main

```

System Administration Log Query Menu

The options on this menu invoke the following:

| <b>Function</b>         | <b>Description</b>   |
|-------------------------|--|
| Log by Time Query Menu  | Invokes the Log by Time Query Menu, where you can view Service logs and Control Record logs.<br><br>Log information is presented in time order, beginning with the most recent information.  |
| Log by Value Query Menu | Invokes the Log by Value Query Menu, where you can view Service logs.<br><br>Values (for example, Spectrum service names) are presented in alphanumeric order, along with log information such as such as system activities, changes to records, messages, dates, and times. |

---

---

## USING THE CONSTRUCT SPECTRUM CONTROL RECORD

The Construct Spectrum control record provides current information about the status of your system. Using the Maintain Control Record window, you can set Entire Net-Work timeout values, enable or disable Natural Security, and specify a security exit.

The following topics are covered:

- **Accessing the Maintain Control Record Window**, page 90
- **Using the Maintain Control Record Window**, page 91
- **Spectrum Control Record Log**, page 94

## Accessing the Maintain Control Record Window

- To access the Maintain Control Record window:
  - 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “MM” in the Function field of the System Administration Main Menu.  
The System Administration Maintenance Menu is displayed.
  - 3 Enter “SC” in the Function field of the System Administration Maintenance Menu.  
The Maintain Control Record window is displayed.



## Modifying Control Data

- To modify the Natural Security setting, the security exit, or the Entire Net-Work timeout setting:
  - 1 Enter “M” in the Action field.
  - 2 Change the value in the appropriate field. For example, you can:
    - Mark the Use Natural Security field to enable Natural Security.
    - Type a new value in the security exit field to change the security exit.
    - Type a new value in the Entire Net-Work timeout field.
  - 3 Press Enter.

The following sections explain the Natural Security, security exit, and Entire Net-Work timeout values.

### Natural Security

When Spectrum is installed, the Natural Security field is not marked (meaning Natural Security is not enabled). To enable Natural Security, mark the Natural Security field.

*Note: Once Natural Security is enabled, all group and user maintenance and query functions are integrated into Natural Security panels and data. An additional query for a user's groups and an additional maintenance option for User/Library link functions is added to the menu structure.*

For more information about how the Construct Spectrum Administration subsystem is affected when Natural Security is enabled, see **Using Natural Security With Construct Spectrum**, page 40.

### Security Exit

By default, this field displays the routine (BS\_USEC2) that is supplied with Construct Spectrum to evaluate the security settings when requests are made to the server data. You can modify and replace the security exit used by your system.

For more information about the security exit, see **Setting Security Options**, page 127.

## Entire Net-Work Timeout

When Spectrum is installed, the default setting for the Entire Net-Work timeout value is “60” (seconds). To change the Entire Net-Work timeout setting, enter the new value in the Entire Net-Work timeout field.

---

**Tip:** This value should be adjusted to match the lowest of the timeout values of all the Entire Net-Work nodes running in your network.

---

## Spectrum Control Record Log

All changes to the Spectrum control record are logged. They are displayed in date and time order on two panels.

### Accessing the Spectrum Control Record Log

- To access the Spectrum Control Record Log:
  - 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “LM” in the Function field of the System Administration Main Menu.  
The System Administration Log Query Menu is displayed.
  - 3 Enter “LT” in the Function field of the System Administration Log Query Menu.  
The System Administration Log by Time Query Menu is displayed.
  - 4 Enter “SC” in the Function field of the System Administration Log by Time Query Menu.  
The first “Query Spectrum Control Record Logs by Time” panel is displayed.

---

## DEFINING GROUPS AND USERS

Essential tasks in setting up a secure environment for a Construct Spectrum application are creating groups to represent collections of users, defining users to the Construct Spectrum Administration subsystem, and then associating users with the appropriate groups. This chapter explains these tasks.

The following topics are covered:

- **Introduction**, page 96
- **Defining Groups**, page 99
- **Defining Users**, page 104
- **Queries and Logs for User and Group Tables**, page 113

For related information, see:

- **Setting Construct Spectrum Security Options**, page 123.  
This chapter explains the next step in setting up a secure environment for Spectrum applications: how to control a group's access to domains, objects, and methods.
- **Creating a Domain and Setting Up Security**, page 52, *Construct Spectrum Programmer's Guide*  
In this chapter you will find information about creating domains.

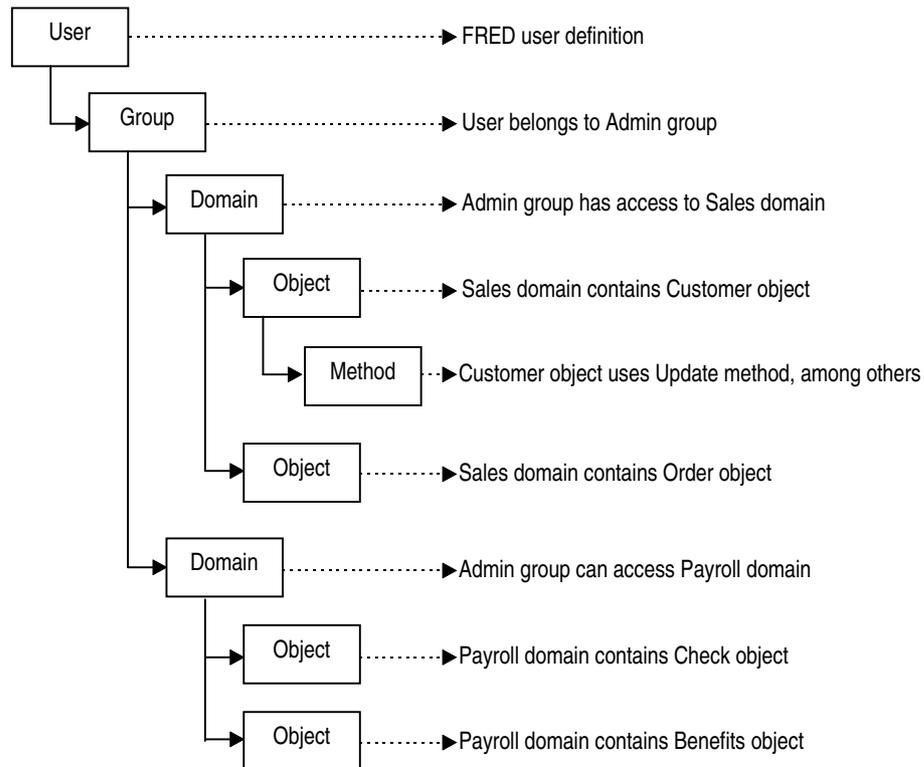
## Introduction

To create a secure environment for a Spectrum application, administrators work with groups, users, domains, objects, and the objects' methods.

- **Groups**  
As an administrator, you will create groups to represent multiple users with the same need and authority to access particular business objects and their methods. Groups usually associate people who perform similar tasks. For example, an organization might have groups for Quality Assurance, Sales, and Management, among others.
- **Users**  
You can then identify individual user IDs to the Spectrum Administration subsystem. Users' profiles include the groups to which they belong.
- **Domains, objects, and methods**  
Domains, like groups, provide a way of collecting similar entities into a convenient unit. In this case, domains collect objects that represent business functions. For example, in the diagram on the next page, the Sales domain encompasses the Customer and Order objects. Each object has one or more methods (such as Update) assigned to it.

The usefulness of groups and domains becomes clear when you define Security settings: with one setting, you can grant (or deny) a group of users access to a domain containing several objects. (On the other hand, you can also set Security privileges for individual objects and methods.) For more information, see **Setting Construct Spectrum Security Options**, page 123.

The following diagram shows the relationships between users, groups, domains, objects, and methods:



#### Relationship Between Users, Groups, Domains, Objects, and Methods

Because the person with the user ID FRED is part of the Admin group, he potentially has access to the Sales and Payroll domains, their objects (such as Customer and Check) and the objects' methods (such as Update). As you will see in **Setting Construct Spectrum Security Options**, page 123, you can grant or deny access privileges on several levels. For example, you can grant or deny FRED access to an entire domain or to an individual object or method.

The rest of this chapter explains how to create groups and users, whether you are using Natural Security or Construct Spectrum's security facilities only. It also explains the queries and logs that are available when you wish to view information about existing groups and users.

## Defining Groups

All users of an application that employs Spectrum services must be known to the Construct Spectrum Administration subsystem through their associations with groups. To access any of the application's objects, a user must be a member of at least one group that is linked to a domain containing the object. A user can belong to a maximum of ten groups.

Groups are used to collect users into larger entities so that Security settings can be defined and checked at a higher level. This eliminates the need to define Security settings for each user. However, it is still possible to define Security privileges for a specific user by creating a group for that user alone.

Construct Spectrum allows users and groups to be defined either within the Spectrum system files or by linking to Natural Security user and group definitions. The following sections explain how to define groups both with and without Natural Security. For more information, see **Using Natural Security With Construct Spectrum**, page 40.

### Defining Groups Using Spectrum Security

Use the Maintain Group Table panel in the Spectrum Administration subsystem to define groups. Each user can belong to one or more of these groups.

#### Accessing the Maintain Group Table Panel

- To access the Maintain Group Table panel:
  - 1 Enter "SA" in the Construct Spectrum Administration Subsystem Main Menu. The System Administration Main Menu is displayed.
  - 2 Enter "MM" in the Function field of the System Administration Main Menu. The System Administration Maintenance Menu is displayed.

- 3 Enter "GR" in the Function field of the System Administration Maintenance Menu. The Maintain Group Table panel is displayed:

```

BSGR_MP          Construct Spectrum Administration Subsystem      BSGR__11
Feb 27          Maintain Group Table                             10:51 AM

Action (A,B,C,D,M,N,P)  _

Group ID.....: _____
Group Name.....: _____

Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
confm help  retrn quit          flip  pref                               main

```

Maintain Group Table Panel

Once you have added the groups you need, you can use the group identifiers when defining users.

## Defining Groups Using Natural Security

When Natural Security is enabled in the Construct Spectrum Administration subsystem, you will access the Maintain Group Profile window instead of the maintenance panel supplied with Construct Spectrum.

## Accessing the Maintain Group Profile Window

- To access the Maintain Group Profile window:
- 1 Enter “SA” in the Construct Spectrum Administration Subsystem Main Menu. The System Administration Main Menu is displayed.
  - 2 Enter “MM” in the Function field of the System Administration Main Menu. The System Administration Maintenance Menu is displayed.
  - 3 Enter “GR” in the Function field of the System Administration Maintenance Menu. The Maintain Group Profile window is displayed. In the following window, the fields have been filled in for an example group called “TEST”.

```

BSSMMENS          Construct Spectrum Administration Subsystem          CDLAYMN1
Jul 08            System Administration Maintenance Menu              03:53 PM

          Functions
+-----Maintain Group Profile (Natural Security)-----+
| Function ..... AD                                           |
| User ID ..... TEST_____|
| User Type ..... G                                           |
| Library ID or copied from user ..... _____|
|
|          Press any PF-key to stop
|
+-----+
.   Terminate
-----+
Function ..... gr

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      help  retrn quit          flip                               main

```

Maintain Group Profile Window

## Using the Maintain Group Profile Window

➤ To use the Maintain Group Profile window:

- 1 In the Function field, type the code for the function you wish to perform. The options are:
  - AD for Add
  - CO for Copy
  - DI for Display
  - MO for Modify

- 2 In the User ID field, type the name of the new or existing group.

*Note: Each group ID must be unique among all user IDs and library IDs defined to Natural Security.*

- 3 In the User type field, type “G” for group.

- 4 Press Enter.

The Add User panel is displayed:

```

14:26:03                *** Natural Security ***                98-03-13
                        - Add User -

User ID ..... TEST                Modified .. 96-12-04 by DEVRE
User Name .... _____
User Type .... G (Group)

Members                      Libraries
-----                      -----
_____                      Default .. _____
_____
_____
_____                      Batch User ID ..... _____

No. members 2

Additional Options ... N

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit  AddOp           Flip                               Canc
  
```

Add User Panel

*Note: You can use the Add User panel to add a group. A group is distinguished from an individual user by its user type: G.*

For more information about using the Add User panel, see the *Natural Security Manual for Mainframes*.

You must define all necessary group and domain combinations using the Maintain Object Security by Group panel, as explained in **Adding a Group and Domain Combination**, page 126.

## Defining Users

Application administrators define users so that their passwords can be checked when they attempt to perform actions in the applications supported by the Construct Spectrum Administration subsystem.

If you are not using Natural Security, you will use the Maintain User Table panel to identify each user to the Construct Spectrum Administration subsystem, specify the user's debug library and file, specify the user's preferred language, and list the groups to which the user belongs.

When Natural Security is enabled, you will use the Maintain User Profile window and the Add User panel to define users. To associate users with groups, you will use the Maintain User's Groups window. You can also link users to protected libraries using the Library Maintenance / Link Users window.

If you are using Natural Security, the user's debug library and filename default to the user's ID. The preferred language is set to English by default.

## Defining Users Using Spectrum Security

Use the Maintain User Table panel in the Spectrum Administration subsystem to define users.

### Accessing the Maintain User Table Panel

- To access the Maintain User Table Panel:
  - 1 Enter "SA" in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter "MM" in the Function field of the Systems Administration Main Menu.  
The System Administration Maintenance Menu is displayed.

- 3 Enter “US” in the Function field of the System Administration Maintenance Menu. The Maintain User Table panel is displayed:

```

BSUS__MP          Construct Spectrum Administration Subsystem      BSUS__11
Feb 27           Maintain User Table                             11:07 AM

Action (A,B,C,D,M,N,P)  _

User ID.....: _____
Password.....: _____

Name.....: _____

Debug Library.....: _____
Debug Filename.....: _ ('T'imestamp; 'U'ser ID)

Preferred Language.....: _

Groups.....: _____ *
              _____
              _____
              _____
              _____

Direct Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--
confm help retrn quit          flip pref                               main

```

### Maintain User Table Panel

The following sections explain debug information and how to associate users with groups.

## Defining a Debug Library and File

On the Maintain User Table panel, you can specify a user’s debug library and file. Using Trace-Option(1), you can store data transmitted between the client and server in the debug file. For more information about Trace-Option(1), see **Debugging Your Client/Server Application**, page 201, *Construct Spectrum Programmer’s Guide*.

The Debug Library value defaults to your user ID.

Use the Debug Filename field to specify how the debug file is named.

| <b>Option</b> | <b>Code</b> | <b>Description</b>   |
|---------------|-------------|--|
| Timestamp     | T           | The current time value is the filename. This allows multiple copies of debug data to exist; later copies do not overwrite earlier ones. Each file is assigned a name consisting of "T" followed by a time value of HHMMSSST. For example, files generated at 12:23:54.7 and 18:12:51.9 would have the names "T1223547" and "T1812519", respectively. |
| User          | U           | The user ID is the filename. This option is the default. Each request from the client overwrites the contents of the debug file.   |

*Note: When a user invokes a subprogram proxy from the Application Administration Main Menu, the default values for the debug library and file are taken from the settings defined in the Maintain User Table panel.*

## Associating Users with Groups

Each user must be associated with at least one group. If a user does not belong to any group, Construct Spectrum will reject all access requests made by the user. When you define Security settings, you will associate groups and domains to specify which users have access to which objects. Therefore, a user must belong to at least one group that has been associated with the domain that contains the objects the user needs to use.

You can specify group(s) when you are adding a new user or modifying the profile of an existing user.

## Defining Users Using Natural Security

When Natural Security is enabled in the Construct Spectrum Administration subsystem, you will access the Maintain User Profile window instead of the maintenance panel supplied with Construct Spectrum.

To associate users with groups, use the Maintain User's Groups window. You can also link users to protected libraries using the Library Maintenance / Link Users window.

The following sections explain how to access the appropriate panels and windows, add a user, associate a user with groups, and link a user to libraries.

### Accessing the Maintain User Profile Window

- To access the Maintain User Profile window:
  - 1 Enter "SA" in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter "MM" in the Function field of the System Administration Main Menu.  
The System Administration Maintenance Menu is displayed.

- 3 Enter "US" in the Function field of the System Administration Maintenance Menu. The Maintain User Profile window is displayed:

```

BSSMMENS          Construct Spectrum Administration Subsystem          CDLAYMN1
Jul 08            System Administration Maintenance Menu                03:58 PM

          Functions
+-----Maintain User Profile (Natural Security)-----+
|
| Function ..... AD
| User ID ..... DEVRE
| User Type ..... P
| Library ID or copied from user .....
|
| Copy with links (CO function only).. N ('Y' or 'N')
|
|          Press any PF-key to stop
|
+-----+
Function ..... US

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit          flip                                main

```

Maintain User Profile Window

### Using the Maintain User Profile Window

- To use the Maintain User Profile window:
- 1 In the Function field, type the appropriate code for the function you wish to perform. The options are:
    - AD for Add
    - CO for Copy
    - DI for Display
    - MO for Modify
  - 2 In the User ID field, type the name of the new or existing user.

*Note: Each user ID must be unique among all user IDs and library IDs defined to Natural Security.*

- 3 Press Enter.  
The Add User panel is displayed:

```

14:29:10                *** Natural Security ***                98-03-13
                        - Add User -

                                Modified .. 98-02-21 by DEVRE
User ID ..... DEVRE
User Name ....
User Type .... P (A=Administrator, P=Person, M=Member)

Privil. Groups          Libraries          Password
-----
_____                Default .. _____                New Password
_____                Last ..... _____                Change after ___ days
_____
_____                ETID                                Batch User ID ..... _____
_____                -----                                Language ..... 1
_____                Default .. _____                Private Library ... Y
No. groups 3          Last ..... _____

Additional Options ... N

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit  AddOp      Flip                                Canc

```

### Add User Panel

For more information about using the Add User panel, see the *Natural Security Manual for Mainframes*.

## Associating Users with Groups

Each user must be associated with at least one group. If a user does not belong to any group, Construct Spectrum will reject all access requests made by the user. When you define Security settings, you will associate groups and domains to specify which users have access to which objects. Therefore, a user must belong to at least one group that has been associated with the domain that contains the objects the user needs to use.

After you have added a user, use the Maintain User's Groups window to associate the user with one or more groups.

### Accessing the Maintain User's Groups Window

- To access the Maintain User's Groups window:
- 1 Enter "SA" in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter "MM" in the Function field of the System Administration Main Menu.  
The System Administration Maintenance Menu is displayed.
  - 3 Enter "UG" in the Function field of the System Administration Maintenance Menu.  
The Maintain User's Groups window is displayed:

```

BSSMMENS          Construct Spectrum Administration Subsystem          CDLAYMN1
Oct 01,98         System Administration Maintenance Menu              03:16 PM
+-----Maintain User's Groups (Natural Security)-----+
|
| Group ID ..... _____
| Add / DElete / LIst members ..... _____
| Start value for list function ..... _____
|
| Members
|
| 1 _____ 2 _____ 3 _____ 4 _____
| 5 _____ 6 _____ 7 _____ 8 _____
| 9 _____ 10 _____ 11 _____ 12 _____
| 13 _____ 14 _____ 15 _____ 16 _____
| 17 _____ 18 _____ 19 _____ 20 _____
| 21 _____ 22 _____ 23 _____ 24 _____
| 25 _____ 26 _____ 27 _____ 28 _____
Function | 29 _____ 30 _____
|
|                                     Press any PF-key to stop
Command |
Enter-PF +-----+ 1--PF12---
          help  retrn quit          flip          main

```

Maintain User's Groups Window

- To add a user to a group:
- 1 In the Group ID field, type the group name.
  - 2 In the next field, type “LI” to list the members of the group.
  - 3 Press Enter.  
The groups members are listed in the Members columns.
  - 4 In the ADD / DElete / LIst members field, type “AD”.
  - 5 In an empty Members field, type the user ID you wish to add.
  - 6 Press Enter to add the user to the group.

### **Linking Users to Libraries**

Use the library link function to associate users with protected libraries. The Construct Spectrum Administration subsystem uses the link definitions found in Natural Security to determine whether or not a user can access a Natural library. Each library must be linked to the user in Natural Security. The libraries include:

- Each protected library in a steplib.
- The current execution library, if protected
- The SYSTEM (or \*STEPLIB library), if protected

### **Accessing the Library Maintenance / Link Users Window**

- To access the Library Maintenance / Link Users window:
- 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “MM” in the Function field of the System Administration Main Menu.  
The System Administration Maintenance Menu is displayed.
  - 3 Enter “LI” in the Function field of the System Administration Maintenance Menu.

The Library Maintenance / Link Users window is displayed:

```

BSSMMENS          Construct Spectrum Administration Subsystem          CDLAYMN1
Jul 08            System Administration Maintenance Menu              03:59 PM

          Functions
+---Library Maintenance / Link Users (Natural Security)---+
|
| Function ..... LK
| Library ID ..... SAMPLE__
| Library/private library ..... L
|
| User ID (for link functions)..... SAMPLE__
|
|          Press any PF-key to stop
|
+-----+
Function ..... LI

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      help  retrn quit          flip                               main

```

### Library Maintenance / Link Users Window

#### Using the Library Maintenance / Link Users Window

- To link a user to a library:
  - 1 In the Library ID field, type the library.
  - 2 In the User ID field, type the user ID.
  - 3 Press Enter to create the link.

## Queries and Logs for User and Group Tables

In the Construct Spectrum Administration subsystem, you can view queries and logs containing information about the user and group tables. The following sections explain the options available when you are using Construct Spectrum Security only and when you are using Natural Security.

For information about accessing queries and logs for other Construct Spectrum Administration components, see the chapters that explain those components.

### Queries and Logs for Spectrum Security

When you are using only Construct Spectrum Security, you can view queries for the group and user tables, and you can view logs of the activities performed on the tables.

## Accessing Query Panels

- To access the group or user table query panels:
- 1 Enter "SA" in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter "QM" in the Function field of the System Administration Main Menu.  
The System Administration Query Menu is displayed:

```

BSSQMENU   ***** Construct Spectrum Administration Subsystem *****   CDLAYMN1
Aug 25     - System Administration Query Menu -                           02:39 PM

          Functions
          -----
          SE   Query Services

          CL   Query Communication Logs

          GR   Query Group Table
          US   Query User Table

          ?   Help
          .   Terminate
          -----
Function ..... _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      help  retrn quit          flip                               main

```

### System Administration Query Menu

- 3 Enter the appropriate function in the Function field of the System Administration Query menu.  
The panel for the function you entered is displayed.

The options on this menu that are relevant to group and user tables invoke query panels that contain the following types of information:

| <b>Function</b>   | <b>Description</b>  |
|-------------------|---|
| Query Group Table | Lists existing groups and their characteristics as defined on the Maintain Group Table panel. |
| Query User Table  | Lists existing users and their characteristics as defined on the Maintain User Table panel.   |

## Accessing Query Log Panels

- To access the Query Log panels for the group and user tables:
  - 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “LM” in the Function field of the System Administration Main Menu.  
The System Administration Log Query Menu is displayed:

```

BSSLMENU          Construct Spectrum Administration Subsystem          CDLAYMN1
Apr 15            System Administration Log Query Menu                04:47 PM

                Functions
                -----
                LT   Log by Time Query Menu
                LV   Log by Value Query Menu

                ?   Help
                .   Terminate
                -----

Function .....  _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit          flip                                main

```

### System Administration Log Query Menu

- 3 Enter either “LT” for Log by Time Query or “LV” for Log by Value Query. The “LT” option presents information in time order, beginning with the most recent information. The “LV” option presents values (for example, group names) in alphanumeric order. Both options show log information, such as actions, dates, and times.

## Queries for Natural Security

When you are using Natural Security, you can view queries for the group and user tables and users' groups.

### Accessing Query Panels

To access any of the query panels:

- 1 Enter "SA" in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
- 2 Enter "QM" in the Function field of the System Administration Main Menu.  
The System Administration Query menu is displayed:

```

BSSQMENS   ***** Construct Spectrum Administration Subsystem *****   CDLAYMN1
Aug 25      - System Administration Query Menu -                           02:41 PM

          Functions
          -----
          SE   Query Services

          CL   Query Communication Logs

          GR   Query Group Table
          US   Query User Table
          UG   Query User's Groups

          ?   Help
          .   Terminate
          -----
Function .....  __

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      help  retrn quit          flip                               main

```

### System Administration Query Menu

- 3 Enter the appropriate function in the Function field of the System Administration Query menu.  
The Natural Security panel for the function you entered is displayed.





## Query User's Groups

When you enter "UG" in the Function field of the System Administration Query menu, the List Groups for a User window is displayed:

```

BSSQMENS          Construct Spectrum Administration Subsystem          CDLAYMN1
Jul 08            System Administration Query Menu                    04:00 PM
+-----List Groups for a User (Natural Security)-----+
|
| User ID ..... _____
| Start group value ..... _____
|
|                               Press any PF-key to stop
|
+-----+
GR   Query Group Table
US   Query User Table
UG   Query User's Groups
?   Help
.   Terminate
-----
Function ..... UG

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      help  retrn quit          flip                               main

```

List Group for a User Window

Enter the user ID to view the groups to which the user belongs. You can use the Start group value field to scroll to a particular group. When you press Enter, the Display User panel is displayed:

```

14:12:56          *** Natural Security SYSTEM ***          98-05-06
                  - Display User -
The following groups contain the user: DEVRE

```

| No. | User ID | User Name    | Default Library | No. of Members | Group Type | No |
|-----|---------|--------------|-----------------|----------------|------------|----|
| 1   | TEST    | Test Group 0 |                 | 2              |            |    |
| 2   | TEST2   | Test Group 2 |                 | 2              |            |    |
| 3   | TEST3   | Test Group 3 |                 | 2              |            |    |

```

-----
Enter no. to be processed: __      Reposition to: _____      Exit: _
-----
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit AddOp      Flip      Canc

```

### Display User Panel



---

## SETTING CONSTRUCT SPECTRUM SECURITY OPTIONS

Chapter 4, **Defining Groups and Users**, explains how to set up security at the system level. This chapter explains how to set up security for applications. The security options you set control users' access to the application's domain, objects, and methods. When the client requests access to data on the server, the Spectrum dispatch service invokes the Spectrum security service, which then executes the security exit. This Natural routine analyzes the security settings to determine if the user is allowed to access the data. If the user has permission, the call proceeds. If the user is not allowed access to the data, a message is returned to the client and the call ends.

This chapter explains how to define security settings and how to use the security exits provided with Construct Spectrum to interpret the security settings.

The following topics are covered:

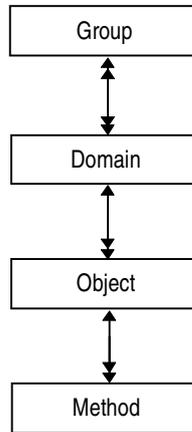
- **Introduction**, page 124
- **Defining Security Settings**, page 125
- **Using the Security Exit Modules**, page 133
- **Using the Cache Synchronization Exit Module**, page 136
- **Queries and Logs for Spectrum Security**, page 137

For related information, see:

- Chapter 2, **Defining and Managing Construct Spectrum Services**, page 47  
This chapter explains how to create a Spectrum security service.
- Chapter 4, **Defining Groups and Users**, page 95  
This chapter explains the relationships between users, groups, domains, objects, and methods.

## Introduction

Spectrum security allows you to define a group's access privileges at multiple levels: domain, object, and method, as shown in the following diagram.



Levels of Security

However, you do not have to define security settings for every object and method. By default, security privileges provided at one level automatically apply to all lower levels in the hierarchy. And different settings can always be explicitly defined for individual items at a lower level. As well, Construct Spectrum provides security exit modules (BS\_USEC\*) that you can modify to evaluate the security settings as needed. Security privileges can be as general or as specific as needed.

At any time, current access privileges can be disabled. This allows you to perform maintenance without changing any of the current security settings. The disabled status is removed by applying the enabled status, which returns the security settings to their prior state.

The version number that is part of an application service definition is ignored in security definitions. Therefore, when you add a new version of an object that is already defined to security, the new version is automatically assigned the same security privileges as older versions.

## Defining Security Settings

This section explains how to access the Maintain Security panels and how to use the panels to set access privileges at several levels.

### Accessing the Maintain Object Security Panels

You can choose between two ways of viewing security information: arranged alpha-numerically either by group or by domain. The following steps explain how to access either option.

- To access the Maintain Object Security panels:
- 1 Enter “AA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The Application Administration Main Menu is displayed.
  - 2 Enter “MM” in the Function field of the Application Administration Main Menu.  
The Application Administration Maintenance Menu is displayed.
  - 3 Do one of the following:
    - Enter “OG” for Maintain Object Security by Group in the Function field of the Application Administration Maintenance Menu. The displayed panel shows groups in alphanumeric order.
    - or
    - Enter “OD” for Maintain Object Security by Domain in the Function field of the Application Administration Maintenance Menu. The displayed panel shows domains in alphanumeric order.

Both panels operate in the same way and offer the same actions. Here is the Maintain Object Security by Group panel:

| BSOSG_MP<br>Apr 17             |        | Construct Spectrum Administration Subsystem<br>Maintain Object Security by Group |                |                 | BSOSG_11<br>9:39 AM |       |
|--------------------------------|--------|--|----------------|-----------------|---------------------|-------|
| Act                            | Group  | Domain   | Current Access | Previous Access |                     |       |
| ---                            | -----  | -----  | -----          | -----           |                     |       |
| -                              | SAMPLE | SAMPLE   | Granted        |                 |                     |       |
| -                              | SPEC01 | SPEC01   | Granted        |                 |                     |       |
| -                              | SPEC02 | SPEC02   | Granted        |                 |                     |       |
| -                              | SPEC03 | SPEC03   | Granted        |                 |                     |       |
| Group: _____ * Domain: _____ * |        |  |                |                 |                     |       |
| Direct command...: _____       |        |  |                |                 |                     |       |
| (PF4=Add)                      |        | Clear  | Disable        | Enable          | Grant               | Purge |
| (PF5=flip)                     |        |  |                |                 |                     |       |

Maintain Object Security by Group Panel

## Adding a Group and Domain Combination

When you add a new group in the Spectrum Administration subsystem, the group will not automatically appear in the Maintain Object Security panel. You must add a group-domain combination in either the Maintain Object Security by Group or Maintain Object Security by Domain panel. The following steps explain how to add a combination in the Maintain Object by Group panel. However, the procedure applies equally in the Maintain Object by Domain panel.

- To add a group and domain combination:
  - 1 In the Group field, select or type the name of the group.
  - 2 In the Domain field, select or type the name of the domain.
  - 3 Press PF4 (Add).  
The new group-domain combination is added to the list on the panel.

Adding a group-domain combination does not automatically give the group access to the domain. You must explicitly set privileges using one of the actions available on the panel. How to access security levels and assign security statuses is explained in the following sections.

*Note: If you add a new object to an existing domain, it automatically inherits the security privileges of the domain-group combination to which it belongs. If a new method is added, it inherits the same security settings as the object to which it belongs.*

## Purging a Group-Domain Combination

The Purge action is only available at the group-domain level. It removes all security statuses for the group-domain combination and its associated objects and methods. While deleting the security definitions, it also deletes all related entries from the security cache. Once all security definitions have been removed, the action also removes the group-domain combination.

## Setting Security Options

This section explains how to set security options at the group-domain, object, and method levels. Although the panels shown are the Maintain Object Security by Group panels, the explanations also apply to the Maintain Object Security by Domain panels.

For information about the security statuses (such as grant or disable) you can assign to group-domain combinations, objects, and methods, see **Security Statuses**, page 131.

### Setting Security Options for Group-Domain Combinations

Settings can be defined for any combination of group and domain. (For more information, see **Adding a Group and Domain Combination**, page 126.) If a setting is defined only at the group-domain level, the group has the same privilege for all objects and methods within the domain.

- To set security options for a group-domain combination:
- Enter the appropriate code in the Act field for the group.  
For information about the statuses, see **Security Statuses**, page 131.

| Act | Group   | Domain   | Current Access | Previous Access |
|-----|---------|----------|----------------|-----------------|
| —   | CST411T | CST411T  | Granted        |                 |
| —   | CST411T | DEMO     | Granted        |                 |
| —   | CST411T | SAGDDEMO | Granted        |                 |
| —   | SYSTEM  | BKR-COMM | Granted        |                 |
| —   | SYSTEM  | DEMO     | Granted        |                 |
| —   | SYSTEM  | SAMPLE   | Granted        |                 |

BSOSG\_MP Jun 11 Construct Spectrum Administration Subsystem  
Maintain Object Security by Group BSOSG\_11 10:21 AM

Group: \_\_\_\_\_ \* Domain: \_\_\_\_\_ \*  
 Command.....: \_\_\_\_\_  
 (PF4=Add) Clear Disable Enable Grant Objects Purge  
 (PF5=flip)

### Maintain Object Security by Group Panel

Use the Clear action if you wish to remove security settings for a group-domain combination. The action also clears the group-domain's settings from the security cache.

### Setting Security Options for Objects

Use the second level in the security hierarchy to allow the group access to specific objects within a domain or to override settings defined at the group-domain level.

➤ To set security options at the object level:

- 1 Enter “O” in the Act field for the group-domain combination.  
A window is displayed showing the objects that belong to the domain. For example:

```

BSOSG_MP          Construct Spectrum Administration Subsystem          BSOSG_11
Jun 11            Maintain Object Security by Group                   10:29 AM

+-----+-----+-----+-----+
| BSOSO_MN          Construct Spectrum Administration Subsystem      |
| Jun 11            Security for Group CST411T/Domain DEMO: Granted  |
|                   10:32 AM                                         |
|-----+-----+-----+-----+
| Act              Object              Current Previous            |
|                |                    Access  Access              |
|-----+-----+-----+-----+
| -  CUSTOMER                                           |
| -  NO-PARMS                                           |
| -  ORDER                                               |
| -  PRODUCT                                             |
| -  SYSCST_CUSPMA                                       |
| -  WAREHOUSE                                           |
|                   *** End of Data ***                    |
| Object:                                                |
| Command.....: _____|
| Clear   Disable  Enable   Grant   Methods   Revoke    |
| (PF5=flip)                                           |
+-----+-----+-----+-----+

```

### Security for Group-Domain Window

Note that the security status for the group-domain combination is displayed in the second window header. The security status applied to the group-domain combination is not shown for each of the domain’s objects in the Current Access field. However, a status defined for the group-domain combination applies to associated objects by default. Only statuses explicitly defined for individual objects are displayed in the Current Access field.

- 2 Enter the appropriate code in the Act field for the object.

For more information about the statuses, see **Security Statuses**, page 131.

Use the Clear action if you wish to remove security settings for an object. The action also clears the object’s settings from the security cache.

## Setting Security Options for Methods

Use the third level in the security hierarchy to allow the group access to specific methods of an object or to override settings defined at the object or group-domain level.

- To set security options at the method level:
- 1 Enter "M" in the object's Act field in the Security for Group-Domain window. The Security for Object window is displayed. For example:

```

BSOSG_MP          Construct Spectrum Administration Subsystem          BSOSG_11
Jun 11            Maintain Object Security by Group                  10:29 AM

+-----+
| BSOSO_MN          Construct Spectrum Administration Subsystem      |
| Jun 11            Security for Group CST411T/Domain DEMO: Granted  |
|                               10:33 AM                             |
+-----+
| BSOSM_MN          Construct Spectrum Administration Subsystem      |
| Jun 11            Security for Object CUSTOMER: None                |
|                               10:33 AM                             |
+-----+
| Act                Method                Current Previous         |
|                   |                     Access  Access         |
+-----+-----+-----+-----+
| _ BROWSE          |                   |                   |                   |
| _ DELETE          |                   |                   |                   |
| _ EXISTS          |                   |                   |                   |
| Method:           |                   |                   |                   |
| Command.....:    |                   |                   |                   |
| Clear            | Disable  Enable  Grant  Revoke  (PF5=flip) |
+-----+-----+-----+-----+

```

### Security for Object Window

Note that the security status for the object is displayed in the second window header. The security status applied to the object is not shown for each of its methods in the Current Access field. However, a status defined for an object applies to associated methods by default. Only a status defined for an individual method is displayed in the Current Access field.

- 2 Enter the appropriate code in the Act field for the method.

For more information about the statuses, see **Security Statuses**, page 131.

Use the Clear action if you wish to remove security settings for a method. The action also clears the method's settings from the security cache.

## Security Statuses

Security statuses can be assigned to any group-domain combination, object, or method in the security hierarchy. To apply the enable, disable, grant, or revoke statuses, enter the appropriate code in the Act field next to the group-domain combination, object, or method, as explained in the previous section.

The following sections explain the generic uses of the statuses. How they are interpreted when a client request is made to the dispatch service depends on the security exit you are using. For more information, see **Using the Security Exit Modules**, page 133.

### No Privileges Status

When a group-domain combination's Current Access field is blank, no privileges have been assigned to the combination. This is the status in effect after you add a new group-domain combination. Privileges can be set for the domain's objects and methods at this point. Or you can set a security option for the combination that will apply to all objects and methods in the domain.

A blank Current Access field has a different meaning when it belongs to an object or method. When you set a group-domain combination's security status, all of its objects and their methods "inherit" that status. However, their Current Access fields will remain blank unless you assign statuses to them individually. Similarly, when you explicitly set an object's security status, that status applies to its methods as well.

## Revoked and Granted Statuses

The granted status allows the group access to the domain, object, or method to which the status is assigned. The revoked status prevents users from accessing the group-domain combination, object, or method to which the status is assigned.

There are different ways to interpret the granted and revoked settings, depending on the security exit you are using. For example, the default security exit (BS\_USEC2) checks access privileges from the lowest level up (from method to group-domain combination) for each group to which a user belongs. It is possible, therefore, for a user to access a method that has the granted status, although the object to which the method belongs has the revoked status.

On the other hand, if you are using the BS\_USEC1 security exit, the revoked status always takes precedence over the granted status. If the group-domain has the status “granted” and the object’s status is “revoked”, the object and its methods are inaccessible. Even if one of the methods of the object has access explicitly granted, the revoked status at the object level will disallow access to any of the object’s methods.

## Disabled and Enabled Statuses

Disabling privileges at any level does not change the current security settings. Rather, it is a temporary override, usually for maintenance purposes. When you apply the disabled status to a group-domain combination, object, or method, the Previous Access field is populated with the value that was in the Current Access field. Once the override is no longer necessary, use the enable action to return the setting to the value in place when the disable status was applied. At any time, the disable status can be changed to grant, revoke, or no privileges.

## Using the Security Exit Modules

Construct Spectrum provides exit modules (BS\_USEC\*) that you can exchange or modify to interpret the security settings differently. These exits receive information from the Spectrum security service when a request is made to the server. The information includes the name of the user, the request being made, the user's groups, and the security settings for the appropriate domain.

The security exit interprets the security settings and returns a flag indicating whether or not to grant access to the user for the request. If the request is not granted, a message number and three substitution values can be returned. (The text associated with the message number is defined in the SYSERR library.) The message data is returned to the client, written to the security log, and placed in the security cache for quicker access on subsequent calls.

The information supplied to the exit is defined in a PDA member called BS\_USECA. This member is supplied in source format specifically to provide context and additional documentation. It should not be changed or recompiled.

To switch security exits, access the Maintain Spectrum Control Record panel and replace the current exit. Then restart all security services. For more information, see Chapter 3, **Using the Construct Spectrum Control Record**, page 89.

The following sections explain the security exits supplied with Construct Spectrum. Note that BS\_USEC2 is the default security exit.

### **BS\_USEC0 (Template)**

This module is a template containing a data section that defines the required parameter data for the security exit. You can insert all the code required to implement security at the application level into this template.

## BS\_USEC1 (“Pessimistic” Checking)

This module implements “pessimistic” security checking. It disallows the user access if any restrictions are found. The algorithm is:

```
If no privileges are defined at any level (domain, object, or method)
then
```

```
    Set the access flag to 'No Privileges'
```

```
    Exit the routine
```

```
End If
```

```
If privileges are disabled at any level then
```

```
    Set the access flag to 'Disabled'
```

```
    Exit the routine
```

```
End If
```

```
If privileges are revoked at any level then
```

```
    Set the access flag to 'Revoked'
```

```
    Exit the routine
```

```
End If
```

```
If privileges are granted at any level then
```

```
    Set the access flag to 'Granted'
```

```
    Exit the routine
```

```
End If
```

```
Evaluate the access flag setting and set the SEC_GRANTED flag
accordingly
```

## BS\_USEC2 (“Optimistic” Checking)

This module is the default security exit. It implements “optimistic” security checking. It allows the user access if any grant status is found for any group at the lowest level in the security hierarchy. The algorithm is:

```
If no privileges are defined (all matrix entries are 0) then
  Set the access flag to 'No Privileges'
  Exit the routine
Else
  For each group do
    If any level (domain, object, or method) is disabled then
      Set the access flag to 'Disabled'
      Go to the next group
    Else
      For each level from 3 to 1 (method to domain) do
        If the current level is revoked
          Set the access flag to 'Revoked'
          Go to the next group
        Else
          Set the access flag to 'Granted'
          Exit the routine
        End If
      End For
    End If
  End For
End If

Evaluate the access flag setting and set the SEC_GRANTED flag
accordingly
```

## Using the Cache Synchronization Exit Module

Whenever security data is updated in the Construct Spectrum Administration subsystem, the affected data in the security cache is cleared. The next request from the client causes the security cache to be populated with up-to-date information.

Whenever object- or method-level security is updated, the cache synchronization exit module (BS\_SYNCN) is called to clear the data from the cache. You can modify it to override the level at which cache cleanup is performed. The source for the exit contains extensive comments to help you modify this routine to your requirements.

Note that you cannot override levels if a change is made to security for a group-domain combination. All entries in the cache for the domain, regardless of group, are deleted.

By default, if a change is made to security at the object level, all entries for the domain-object combination are deleted from the cache. You can modify the exit to specify that all entries for the domain are to be deleted, regardless of object.

By default, if a change is made to security at the method level, all entries for the domain-object-method combination are deleted from the cache. You can modify the exit to specify that all entries for the domain or domain-object are to be deleted.

*Note: Do not specify that deletion start at a lower level than that being updated. For example, if an object's security is updated, do not specify deletion to occur at the method level. If you do so or if no synchronization level is specified, the system will default to the security level being updated.*

For more information about security cache synchronization, see **Synchronizing the Security Cache and Cleanup Queue**, page 141.

## Queries and Logs for Spectrum Security

In the Construct Spectrum Administration subsystem, you can view queries and logs containing information about Spectrum security settings.

For information about accessing queries and logs for other Construct Spectrum Administration components, see the chapters that explain those components.

### Accessing Query and Log Panels

- To access the query and log panels:
  - 1 Enter “AA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The Application Administration Main Menu is displayed.
  - 2 Enter “QM” in the Function field of the Application Administration Main Menu.  
The Application Administration Query Menu is displayed:

```

BSAQMENU          Construct Spectrum Administration Subsystem          CDLAYMN1
Mar 17            Application Administration Query Menu          02:38 PM

                Functions
                -----
                AS  Query Application Service Definitions

                OG  Query Object Security by Group
                OD  Query Object Security by Domain
                SC  Query Security Logs

                DO  Query Domain Table
                ST  Query Steplib Table

                ?  Help
                .  Terminate
                -----
Function ..... _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--
      help  retrn quit          flip          main

```

Application Administration Query Menu

- 3 Enter the appropriate function in the Function field of the Application Administration Query menu.

The query panel for the function you entered is displayed.

The options on this menu for security information invoke query panels that contain the following types of information:

| <b>Function</b>                 | <b>Description</b>   |
|---------------------------------|--|
| Query Object Security by Group  | Lists groups in alphanumeric order, as well as their domains, objects, methods, and security settings.                                       |
| Query Object Security by Domain | Lists domains in alphanumeric order, as well as the groups, objects, methods, and security settings.   |
| Query Security Logs             | Lists security messages generated in response to errors. Messages are listed in date and time order, beginning with the most recent message. |

---

---

## USING CONSTRUCT SPECTRUM TOOLS

Construct Spectrum provides several tools for managing applications and client/server communications. This chapter explains how to use the tools in the Construct Spectrum Administration subsystem and on the client.

The following topics are covered:

- **System Administration Tools**, page 140
- **Client Tools**, page 143

## System Administration Tools

There are several functions included with the Construct Spectrum Administration subsystem that allow you to maintain the system and its data.

### Accessing the System Administration Tools

You can access the tools explained in this section on the System Administration Functions menu.

- To access the System Administration Functions menu:
  - 1 Enter "SA" in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter "SF" in the Function field of the System Administration Main Menu.  
The System Administration Functions Menu is displayed:

```

BSSAMENU   ***** Construct Spectrum Administration Subsystem *****   CDLAYMN1
Aug 25     - System Administration Functions Menu -                       03:18 PM

          Functions
          -----
          SQ   Synchronize security Cache & Cleanup Queue
          SC   Reset security Cache

          ?   Help
          .   Terminate
          -----
Function ..... _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit          flip                               main

```

System Administration Functions Menu

The following sections explain the functions and how to use them.

## Synchronizing the Security Cache and Cleanup Queue

This section explains the purpose of the security cache and cleanup queue and when to use the Synchronize Security Cache & Cleanup Queue function.

### Security Cache

When a Spectrum dispatch service receives a request from the client, the security service assigned to it checks the user's security privileges. If, for every request that is handled by the Spectrum dispatch service, lookups had to be made to a user table, then to a group table, then to an application service definition, and then to a security table, the time required to process a request would be inordinately long. To optimize this process, a cache of security data is built by a security service whenever it first checks the security for a given request. Thereafter, only one table lookup is required to check the access rights of a user to requested data, thereby greatly improving response time after the initial request.

Whenever data that affects the contents of the security cache is updated in the Construct Spectrum Administration subsystem, the affected data in the security cache is cleared. For example, if a subprogram proxy module is regenerated using Natural Construct or if an administrator revokes access privileges for a group, the appropriate data in the security cache is cleared. The next request from the client causes the security cache to be populated with up-to-date information.

### Cleanup Queue

If an abend or other interruption occurs while the security cache is being updated, the update could fail. The result would be that the security cache is not synchronized with the Construct Spectrum Administration subsystem files. The cleanup queue prevents this from happening.

Before an update is made to the security cache, Construct Spectrum writes a record to the cleanup queue containing information about the impending update. Then, if the update fails, the record remains in the queue so that you can synchronize the

security cache with the Construct Spectrum Administration subsystem files. If the update completes successfully, the record in the cleanup queue is automatically deleted.

### **Using the Synchronize Security Cache & Cleanup Queue Function**

Use the Synchronize security Cache & Cleanup Queue function to complete any cache cleanup tasks that failed prior to completion. Typically, you will use this function if clients performing remote CALLNATs receive the error “SPE5065> Request cannot be handled while cache cleanup is in progress” repeatedly. Since the cleanup record is only deleted after the operation completes successfully, this function can be executed at any time and as often as necessary.

## **Resetting the Security Cache**

Use this function to completely clear all entries from the security cache. By emptying the cache contents, this function forces the security service to rebuild all security cache entries.

After specifying your request to clear the cache, an additional choice may be presented. If the cache contains more than 1000 records, you must choose from the following options:

- Submit the cache delete as a batch job. (Recommended, especially for large caches.)
- Run the cache delete online. (This may take a long time if the cache is large.)

---

**Tip:** Use the Reset security Cache function after changing from using the Spectrum Administration subsystem security files (User, Group) to using Natural Security data directly, or vice versa. It clears the cache and allows the cache to be rebuilt based on the new definition of user and group data.

---

*Note: If you change the security mode in a dispatch service or if you change the SAF resource profiles, reset the security cache to clear out the old settings.*

## Client Tools

This section describes the client tools that come with Construct Spectrum. The client tools are:

| <b>Tools</b>             | <b>Description</b>   |
|--------------------------|--|
| Spectrum Service Manager | Administrators and application developers use this program to configure the client to access Spectrum dispatch services running on one or more servers.                            |
| Change Password          | Users use this program to change their Construct Spectrum password on the server.  |
| Remote Clock             | The Remote Clock displays the current time on the server. The program can be implemented in either of two ways: using the Timestamp service or using the Spectrum Dispatch Client. |
| Broker Driver            | The Broker Driver program allows you to interactively make calls such as Send, Receive, and Register.  |

### Spectrum Service Manager

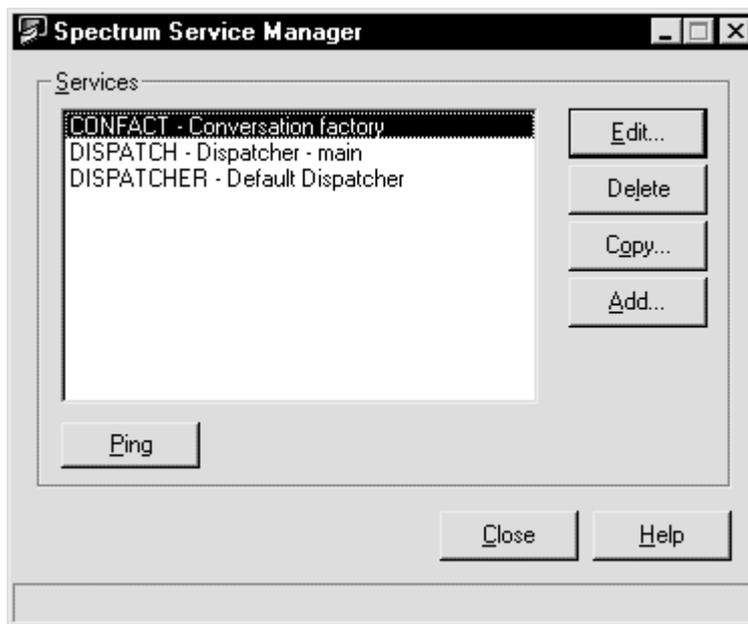
In order for the client to communicate with a Spectrum service running on a server in the network, the client needs to know how the service identifies itself to Entire Broker. This information includes the Broker ID, server class, server name, and service and is stored in a special file called SDC.INI in the Windows directory on each PC. The Spectrum Service Manager is a tool for editing the contents of this file.

You can use the Spectrum Service Manager program to do the following:

- Define the Spectrum services that you would like the client to access.
- Ping Spectrum services to determine if they are running and to test the communications path between the client and server. The ping function checks that DLLs required by Construct Spectrum are installed and in the path. They are checked in this order:
  - BROKERV.B.DLL(from ETB/BIN)

– CDED32.DLL(from Windows\System)

The Spectrum Service Manager looks like this:



Spectrum Service Manager

The Spectrum installation program always installs one Spectrum dispatch service definition, called “DISPATCH”.

Click **Ping** to send a message to a Spectrum service to test the communications path between the client and server. If you receive the message “PING of Service <name> was successful,” the communications path is working. Communication problems are displayed in the status line at the bottom of the dialog box or in a message window.

If you receive the error ETB02150148, indicating that Entire Broker is not active, a possible cause may be 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.

To resolve the error, 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.

Click **Edit**, **Copy**, or **Add** to invoke the **Service Properties** dialog explained in the following section.

### **Service Properties Dialog**

When you edit, copy, or add a Spectrum service definition, you will see the following window:

The screenshot shows a dialog box titled "Service Properties". It contains the following fields and controls:

- Service ID: DISPATCHER
- Name: Default Dispatcher
- Description: (empty text area)
- Entire Broker Parameters section:
  - Broker ID: BKR057
  - Server Class: SPECTRUM
  - Server Name: SYSTEM-SERVICES
  - Service: DISPATCH
- Service type: Dispatcher (411/412)
- Response time-out (seconds): (default)
- Security mode: Spectrum/NSC

Buttons: OK, Cancel, Help

Service Properties

The **Service ID** text box is used to specify a unique name by which a service is known to Spectrum on the client.

Use the **Name** text box to identify the Spectrum service in Construct Spectrum applications. We recommend that you use the same name as defined for the Spectrum service in the Construct Spectrum Administration subsystem to avoid confusion. If you have more than one Spectrum service definition, the name should uniquely identify each one and help determine where that Spectrum service is running or what its primary role is. For example, you could have Spectrum services with names such as "Production Dispatcher," "Development Dispatcher," etc.

Filling in the **Description** text box is optional. The description is used only in the Spectrum Service Manager for comments about the Spectrum service.

The **Entire Broker Parameters** define the Broker ID, server class, server name, and service that uniquely identify a Spectrum service to Entire Broker. These settings must match the settings in the Spectrum Administration subsystem, as shown below:

```

SPBSMD      ***** Construct Spectrum Administration Subsystem *****      SPBSMD11
Aug 25      - Maintain Services -                                          1 more >

  Action (A,B,C,D,M,N,P)  _ Name: DISPATCHER_____
                        Desc: Dispatcher (Production)_____
+- Entire Broker Service Settings -----+
| Broker ID.....: BKR057_____
| Server class.....: SPECTRUM_____
| Server name.....: DISPATCH_____
| Service.....: MAIN_____
| User ID.....: _____
| Password.....: _____
| Attach Service name.....: ATTACH_____ * Attach...: X
+-----+
+- Service Start Parameters -----+
> ATTACH=ATTACH, TRANSACTION=NATBTCH2,
> NATPARAM='FNAT=(19,26), PROFILE=SYSSPECD'
>
| Service start routine...: SUBTASKB *          Source S421      Object S421
+-----+
Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
confm help retrn quit          flip pref          left right main

```

### Maintain Services Panel 1

Use the **Service type** text box to specify the type of service. The following types are available:

| Service type         | Description   |
|----------------------|---|
| Dispatcher           | Spectrum dispatch service (Version 4.2.1 or higher).  |
| Dispatcher (411/412) | Use this setting to indicate that this is a Spectrum version 4.1.1 or 4.1.2 dispatch service. These older dispatch services use a slightly different ping protocol. |

| <b>Service type</b>  | <b>Description (continued)</b>   |
|----------------------|--|
| Attach Server        | Spectrum attach service  |
| Conversation Factory | Spectrum conversation factory  |
| Other                | Any other type of Spectrum service that you may want to ping, such as security servers or your own servers |

If you are viewing a dispatch service, the following information also appears in the Service Properties dialog.

The **Response time-out** text box specifies how long the Spectrum Dispatch Client will wait for a response from the Spectrum service before returning with a “Server is not responding” message. The default value is 15 seconds. You can set it to be any value from 1 to 60 seconds.

**Security mode** specifies the security system the dispatcher is using. This setting must match the service’s SECURITY-MODE parameter, as defined in the Construct Spectrum Administration subsystem. For example:

```

SPBSMD      ***** Construct Spectrum Administration Subsystem *****      SPBSMD11
Oct 13,98   - Maintain Services -                                          1 more >

  Action (A,B,C,D,M,N,P)  _ Name: DISPATCH_____
                          Desc: Spectrum Dispatch Server_____
+- Entire Broker Service Settings -----+
| Broker ID.....: BKR059_____
| Server class.....: SPECTRUMDEV_____
| Server name.....: DISPATCH_____
| Service.....: MAIN_____
| User ID.....: SPSDISP_
| Password.....:
| Attach Service name.....: ATTACH_____ * Attach...: X
+-----+
+- Service Start Parameters -----+
|> NATPARM='FNAT=(19,30),PROFILE=S421BTCH,
|> STACK=(LOGON S421;START)'
|> SECURITY-MODE=SAF
| Service start routine.... SUBTASKB *      Source S421      Object S421
+-----+
Command:
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
confm help  retrn quit      flip pref      left right main

```

### Maintain Services Panel 2

**Tip:** If you are distributing a Construct Spectrum application on multiple clients, you can copy the Spectrum service definitions to other PCs. The definitions are stored in the SDC.INI file in the Windows or Windows NT directory.

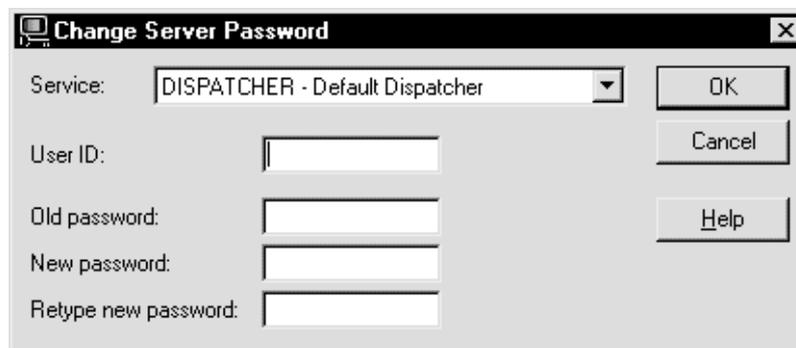
## Change Password Program

*Note:* You cannot use the Change Password program to change your password if you have configured Spectrum to use Natural Security.

Each Construct Spectrum user (developer, administrator, user, etc.) has a user ID and password defined in the Spectrum Administration subsystem. Depending on your network configuration, this user ID and password may or may not be the same as that used to access Windows, your local area network, or other server resources.

The Change Password program allows users to change their Construct Spectrum password without accessing the Construct Spectrum Administration subsystem.

The Change Password window looks like this:

The image shows a Windows-style dialog box titled "Change Server Password". It has a standard title bar with a close button (X) in the top right corner. The dialog contains several input fields and buttons. On the left side, there are four rows of labels and input boxes: "Service:" with a drop-down menu showing "DISPATCHER - Default Dispatcher"; "User ID:" with a text box; "Old password:" with a text box; "New password:" with a text box; and "Retype new password:" with a text box. On the right side, there are three buttons: "OK", "Cancel", and "Help".

Change Server Password

- To change your password:
- 1 Select a Spectrum service in the drop-down list.
  - 2 Type your user ID in the **User ID** text box.
  - 3 Type your current password in the **Old password** text box.
  - 4 Type your new password in both the **New password** and **Retype new password** text boxes.
  - 5 Click **OK**.  
If your password was changed successfully, you will see the message "Password changed successfully."

## Remote Clock Program

The Remote Clock program displays a running clock showing the current time of day on the server. It obtains the server's current time of day using one of two methods:

- By calling a user service, called the `TIMESTAMP` service, which is started from the Construct Spectrum Administration subsystem. This service returns the value of the Natural system variable `*TIMESTAMP`.
- By using the Spectrum Dispatch Client to do a remote `CallNat` to a simple Natural subprogram. This subprogram also returns the value of `*TIMESTAMP`.

The Remote Clock program converts the `*TIMESTAMP` value into the current date and time and then formats it for display. The following diagram shows what the Remote Clock looks like when running:



Remote Clock

## Using the `TIMESTAMP` Service

The `TIMESTAMP` service is a Natural program that illustrates the basics of writing user-defined Spectrum services that can be defined and controlled with the Construct Spectrum Administration subsystem. The Spectrum dispatch service and Spectrum security service are more complex services, but they have essentially the same general structure as the `TIMESTAMP` service.

When the `TIMESTAMP` service starts, it registers a service with Entire Broker. It then enters a receive loop waiting for requests from a client application. It responds to ping and shutdown messages from the Construct Spectrum Administration subsystem. When it receives a shutdown message, it deregisters its service from Entire Broker and then ends. Much of this functionality is encapsulated in the reusable routine `SPUETB`.

The client application that communicates with the `TIMESTAMP` service is the Remote Clock program. This program and the `TIMESTAMP` service use direct calls to communicate with each other. Thus, the round-trip time from client to server and back to client is very short.

## Using the Spectrum Dispatch Client

The Remote Clock program can also use the Spectrum Dispatch Client to do a remote `CallNat`. In this case, it calls the following subprogram:

```
DEFINE DATA
  PARAMETER
    01 #TIMESTAMP (A13)
  LOCAL
    01 #LOCAL-TIMESTAMP (A16)
END-DEFINE
MOVE EDITED *TIMESTAMP (EM=H(8)) TO #LOCAL-TIMESTAMP
#TIMESTAMP := #LOCAL-TIMESTAMP
END
```

This subprogram converts the `*TIMESTAMP` value into a sequence of 16 hex digits, of which the first 13 are significant in Natural. These 13 hex digits are returned in the output parameter `#TIMESTAMP`.

When using this method to obtain the `*TIMESTAMP` value, Construct Spectrum takes care of all of the calls and data marshaling so that the client application and the subprogram it calls can be simple.

## Command Line Parameters

The Remote Clock program accepts a series of command line parameters to control its operation. The full syntax of the command line is:

```
RemoteClock.exe [options]
```

where options can be one or more of the following:

| Option                     | Description   |
|----------------------------|---|
| -s                         | Use the Spectrum Dispatch Client instead of direct calls. The default is to use direct calls.   |
|                            | <i>Note: The options -l, -c, -e, -d, -u, and -p are only valid when the option -s is also used; otherwise they are ignored.</i>   |
| -l                         | Display performance measurements.   |
| -c                         | Use compression. The default is no compression.   |
| -e                         | Use encryption. The default is no encryption.   |
| -d <i>dispatch-service</i> | Use this dispatch service definition from the Spectrum Service Manager. The default is to use the first definition in the SDC.ini file.   |
| -t <i>title</i>            | Display this title in the window title. The title cannot include spaces.  |
| -a <i>time-adjustment</i>  | Adjusts the time returned by the server. The time-adjustment has the following format:<br><i>dDhHmMsStT</i><br>where <i>d</i> , <i>h</i> , <i>m</i> , <i>s</i> , and <i>t</i> specify positive or negative units of days, hours, minutes, seconds, or tenths-of-a-second, respectively. For example:<br><code>RemoteClock -a -2d6h</code><br>adjusts the time returned by subtracting two days and adding six hours.<br>You would use this option if the value returned by *TIMESTAMP has not been adjusted to your local time zone, for example. |
| -u <i>user-ID</i>          | Use this user ID to perform remote CallNats.  |
| -p <i>password</i>         | Use this password to perform remote CallNats.   |

## Broker Driver Program

The Broker Driver program allows you to interactively make calls such as Send, Receive, and Register. This program is invaluable when you are developing client or server applications and would like to test how a particular function works or how behaves with a certain set of input parameters.

*Note: The Broker Driver program always sets API-TYPE to 1 and API-VERSION to 2. Thus, you cannot use it to test version 3 calls such as unit of work.*

The Broker Driver program looks like this:

**Broker Driver**

Broker ID:   Automatic logon   
User ID:   
Token:   
Environment:  Security Token:

Function:  API Version:  2  3  
Service:   
Conversation ID:   
Send Data:   
Wait:  Client User ID:   
Option:  Conversation Status:   
Receive Length:  Return Length:   
User Data:   Convert from EBCDIC  Display Hex  
Receive Data:

Unit of Work  
ID:  User Data:   
Store:  AD Count:   
Time:  Status:   
Status Persist:

Last Broker Message:  
Round-trip Time:

Broker Driver

If you are familiar with the ACI, you will recognize many of the fields on the dialog box. Only a few text boxes require explanation:

- The **Automatic logon** check box is used when you have EntireX security installed. When this check box is selected and a security error occurs, such as an invalid user ID or password, the Broker Driver program displays a logon dialog that lets you enter your user ID, password, and optionally your new password. If this check box is not selected, you should enter the user ID, password, and new password into the fields provided.
- The **Security token** text box shows the contents of the security-TOKEN field in the Broker control block. Because this field is binary, the contents are displayed as a series of hex pairs.
- The **Service** text box is used to enter the SERVER-CLASS, SERVER-NAME, and SERVICE values in the Broker control block. Enter these three values in this order, with a slash character between values. For example: SPECTRUM/DISPATCH/CMD. The **Service** text box also contains a drop-down list, and it remembers all of the values entered into this field. Retrieve a previous value by selecting it from the drop-down list.
- Enter a conversation ID into the **Conversation ID** text box or choose one of the predefined values: NONE, NEW, ANY, or OLD.

After entering all of the values for a Broker call, click **Go!** to execute the call. After the call completes, the error class, error number, and error text will be displayed at the bottom of the dialog box, along with the total round trip time in milliseconds. If any of the Broker control block fields were updated by the call (such as the conversation ID), the corresponding fields on the dialog box will show those values.

---

## DEPLOYING THE CONSTRUCT SPECTRUM ADMINISTRATION SUBSYSTEM

This chapter explains the steps involved in deploying the Construct Spectrum Administration subsystem in development and production environments.

The following topics are covered:

- **Introduction**, page 158
- **Example Deployment Scenario**, page 159
- **Creating the Example Deployment Scenario**, page 161
- **Data Transfer Utilities**, page 165
- **Layout of the Export Users and Groups Workfile**, page 175

For information about deploying a Construct Spectrum application, see **Deploying Your Client/Server Application**, page 237, *Construct Spectrum Programmer's Guide*.

## Introduction

Deploying a Construct Spectrum application is similar to distributing a traditional Natural application. However, the Construct Spectrum runtime environment must be installed on each machine. This installation includes:

- All modules related to the Construct Spectrum Administration subsystem
- All modules related to the Construct Spectrum dispatch, security, and attach services
- One or more copies of the system files

Each runtime environment also requires at least one dispatch, security, and attach service.

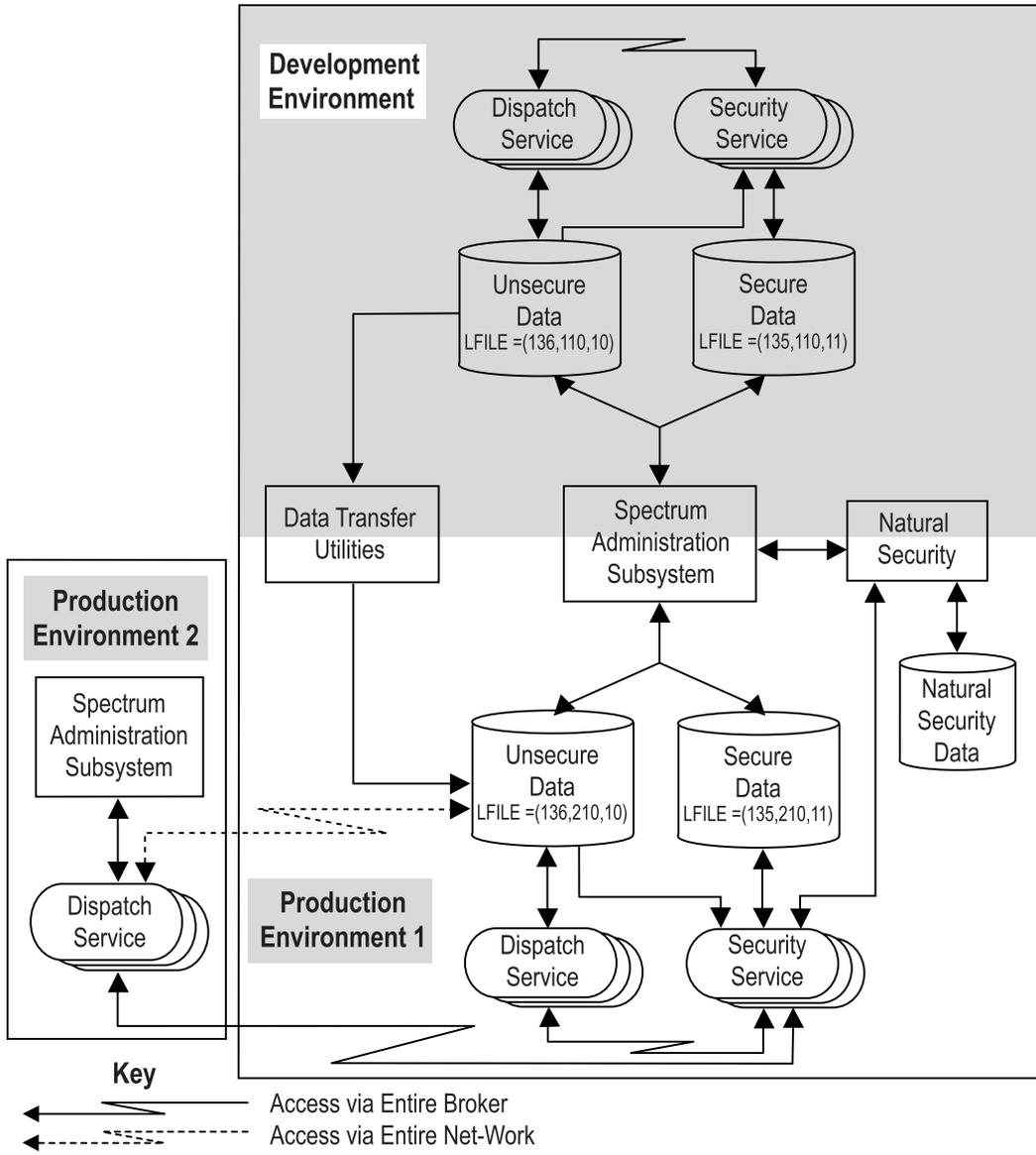
Each runtime environment accesses the same centrally-located data by means of direct calls through one of the following:

- Entire Net-Work  
Or
- Entire Broker, if it is configured to use TCP/IP as the network transport protocol  
Or
- A Spectrum security service, in the case of secure data.

## Example Deployment Scenario

The following diagram shows a scenario with one development and two production environments, one of which is on a remote machine. The Construct Spectrum Administration subsystem is installed on both machines, and Production environment 1 contains a copy of the Construct Spectrum system files. In this scenario, Entire Net-Work is installed on both machines. Entire Broker is installed on the machine with both the Development environment and Production environment 1.

The example scenario shows the option of having Natural Security installed on the machine that houses the Development environment and Production environment 1 to control users' access to secure data. To use Natural Security on the remote machine in the scenario, another copy of Natural Security would have to be installed on that machine, a security service would be needed, and both copies of the Natural Security software would have to be configured to update the Spectrum system files with security information.



Example Deployment Environment

## Creating the Example Deployment Scenario

This section explains the prerequisites and tasks involved in setting up the example scenario illustrated on page 159. Refer to these steps when planning your own deployment strategy.

### Step 1 — Ensure that the Production Environments have the Prerequisite Software Installed

- Entire Broker V2.1 or later must be installed on one or more of the machines. In the example scenario on page 159, it makes sense to install Entire Broker on the machine that contains both development and production environments.
- Either Entire Net-Work Client V2.4.1 (or later) must be installed on every PC that will use the Spectrum application, or EntireX V4.2.1 must be installed and configured to use TCP/IP as the network transport protocol.
- If you are not using EntireX configured for TCP/IP, Entire Net-Work V5.5.1 or later must be installed on each server machine.
- Natural V2.3.2 or later must be installed on each server machine.
- ADABAS V6.2 or later must be installed on each server machine.

For information about installing and configuring these products, see their installation manuals.

### Step 2 — Ensure that Construct Spectrum is Installed on Each Machine

The Construct Spectrum Administration subsystem is used to define and manage Spectrum services and maintain Construct Spectrum Administration subsystem data. You can access a full set of queries and log data in the subsystem.

Construct Spectrum must be installed on each computer. The example scenario on page 159 shows the Construct Spectrum Administration subsystem installed on the machine that hosts the Development environment and Production environment 1 and on the remote computer that houses Production environment 2.

For information about installing Construct Spectrum, refer to the *Construct Spectrum and SDK Mainframe Installation* and *Construct Spectrum and SDK Client Installation*.

### **Step 3 — Copy Construct Spectrum System Files**

In the example deployment scenario, Production environment 1 shares the Spectrum Administration subsystem with the Development environment. While it is possible to share the Spectrum system files between development and production environments, most organizations will want to create separate copies of these files to better ensure the integrity of their production data.

The files are accessed through logical file numbers, and they can be installed using any available physical file numbers. You can assign these logical file settings dynamically using the LFILE parameter, or you can link them into the Natural nucleus using the NTFILE parameter. Both files must be accessible from the Spectrum dispatch service, the Spectrum security service, and the Construct Spectrum Administration subsystem.

#### **Logical File 135**

Logical file 135 contains all information concerning users, their security privileges, and the security cache. This information is stored separately from other dispatcher information to allow this sensitive data to be enciphered, if desired.

#### **Logical File 136**

Logical file 136 contains all information used by Construct Spectrum except for the user and security information stored on logical file 135.

### **Step 4 — Ensure that the Spectrum System File Database is set up in Entire Net-Work (Optional)**

If the environment uses Entire Net-Work for its network transport protocol, make sure that the network administrator has set up the Spectrum system file database so that it is accessible to all appropriate clients.

## Step 5 — Define and Start an Attach Service

At runtime, an attach service is used to launch other services as they are needed. If possible, you should configure your system so that an attach server is started as part of the operating system activation process.

## Step 6 — Define and Start Dispatch and Security Services

At runtime, the dispatch service is responsible for invoking the Natural application services requested by the client. This service accesses the Construct Spectrum system files for self-configuration on startup and for writing messages. It also uses the Spectrum security service (defined as one of its remote services) to determine the validity and location of the requested target service.

The dispatch service communicates with the security service to validate users and check their privileges and requests for application services. At least one Spectrum security service must be available to an environment. It can be installed on any machine that has access to the Construct Spectrum Administration subsystem data. However, to minimize network traffic, the Spectrum security service should run in the same location as the Construct Spectrum Administration subsystem data.

Normally, dispatch and security services are started up automatically on demand by an attach service.

To learn how to define Spectrum services, see **Defining and Managing Construct Spectrum Services**, page 47.

### Natural Security

Spectrum security services can be configured to work with Natural Security so that you can use Natural Security users and groups instead of Spectrum's tables. If you are using Natural Security, the dispatch and security services must have access to Natural Security data. When Natural Security is operating, the dispatcher makes calls to Natural Security to help determine client authorizations, and the dispatch service "impersonates" the client when executing application service requests.

## Step 7 — Transfer Domains

Decide which applications you wish to make available in the production environments and identify their domains.

To export and import domains, use the Domain Transfer Utilities in the Construct Spectrum Administration subsystem. When you use the utilities to copy the table data from one environment to another, the domains, application service definitions and, optionally, the steplib chains are also copied.

For information about importing and exporting domains, see **Transferring Domains**, page 165.

## Step 8 — Transfer Group and User Tables (Optional)

If users and groups have already been defined in the production environments, omit this step. Otherwise, use the Export Groups and Import Groups utilities to copy this data to your production environments.

For more information about exporting and importing groups, see **Transferring Groups**, page 169.

If you use the Export and Import Groups utilities, the permissions that grant access between groups and domains are not transferred. Therefore, you must set security definitions in the Construct Spectrum Administration subsystem for the applications you are distributing.

For more information, see **Setting Construct Spectrum Security Options**, page 123.

*Note: If you wish to make a mirror image copy of the Spectrum system files, use ADAULD to unload and ADALOD to reload.*

## Data Transfer Utilities

Using the data transfer utilities, you can copy domains and groups between one Spectrum system file and another.

The Domains and Groups Export utilities copy data either to a workfile or a PC file, depending on the definition of the WORK parameter in your NATPARM. Similarly, the Domains and Groups Import utilities load data either from a workfile or PC file, depending on the definition of the WORK parameter in your NATPARM.

The following sections explain how to use the data transfer utilities from the Construct Spectrum Administration subsystem. For information about using the utilities in batch, see **Using the Data Transfer Utilities in Batch Mode**, page 173.

If you are transferring data between version 4.1.1 and 4.2.1 Administration subsystem files, invoke the utilities directly, rather than using the Administration subsystem menus. (If you use the menus to perform the tasks, you will have problems with the Product License system.) Use the following commands at the NEXT prompt to invoke the utilities' windows directly:

| <b>Utility</b>         | <b>Command</b> |
|------------------------|----------------|
| Export Domains utility | BS_EXDOM       |
| Import Domains utility | BS_IMDOM       |
| Export Groups utility  | BS_EXGRP       |
| Import Groups utility  | BS_IMGRP       |

### Transferring Domains

Use the Export Domains and Import Domains utilities to transfer domains from one Construct Spectrum Administration subsystem file to another. The Export Domains utility copies the following information to a data file:

- Domain definitions
- All of the application service definitions to which the domain is attached. This information includes the object descriptions, methods, and subprogram proxy names

attached to the application service definition

- All of the steplib chains attached to the domain

## Accessing the Domain Transfer Utilities

➤ To access the Domain transfer utilities:

- 1 Enter “AA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The Application Administration Main Menu is displayed.
- 2 Enter “MM” in the Function field of the Application Administration Main Menu.  
The Application Administration Maintenance Menu is displayed.
- 3 Enter “DT” in the Function field of the Application Administration Maintenance Menu.  
The Transfer Domains Menu is displayed:

```

BS_XFER2   ***** Construct Spectrum Administration Subsystem *****   CDLAYMN1
Mar 20                - Transfer Domains Menu -                               10:51 AM

                Functions
                -----
                XD   Export Domains
                ID   Import Domains

                ?   Help
                .   Terminate
                -----
Function .....  _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      help  retrn quit          flip                                main

```

### Transfer Domains Menu

- 4 Enter the code in the Function field for the utility you would like to access.  
The window for the utility appears.

## Exporting Domains

- To export domains:
- 1 Enter “XD” in the Function field of the Transfer Domains Menu.  
The Export Domains window appears:

```
Export Domains
BS_EXDOM          BSEXDOM0

Domains
From.....
To.....
Enter-PF1---PF2---PF3---PF4---PF5-
confm help  retrn quit  deflt flip
```

### Export Domains

- 2 Specify the domain or range of domains to be exported. You have the following options:
  - To export one domain, type it in the From field.
  - To export a range of domains, type the beginning and ending values in the From and To fields, respectively.
  - To export all domains with the same initial characters, type the characters followed by an asterisk (\*) in the From field.
  - To export all domains up to and including a specific domain, type the domain in the To field.
- 3 Press Enter to start the export process.  
If you are exporting to a PC file, the Download dialog appears, in which you can specify the file into which the data will be loaded.

When the data is exported, a message informs you of its successful completion. If the export operation was not successful, a message explains the problem.

## Importing Domains

- To import domains:
  - 1 Log on to the library to which you wish to import the domains and invoke the Construct Spectrum Administration Subsystem.
  - 2 Access the Transfer Domains Menu.
  - 3 Enter “ID” in the Function field of the Transfer Domains Menu.  
The Import Domains window appears:

```

BS_IMDOM                                Import Domains                                BSIMDOM0

      Import Domains.....: X
      <OR>
      Only scan input file and generate report.: _

      Replace steplib chains.....: X
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---
      help  retrn quit          flip
  
```

### Import Domains

- 4 Specify the options you wish to use:
  - To import data to the current library, mark Import Domains.
  - If you wish to only create a report of the data to be imported, mark the “Only scan input file and generate report” field. The report lists the steplib chains, domains, and interfaces (application service definitions). The report is written to the default device for Report 1, for example a printer, set up by your system administrator.
  - The “Replace steplib chains” field is marked by default, meaning that steplib chains in the target environment with the same names as those being imported will be replaced by the newcomers. If you do not wish to replace the steplib chains, delete the “X” in this field.
- 5 Press Enter.  
If your NATPARM specifies a PC file, the Upload Data dialog appears, in which you can specify the file to be uploaded or scanned.

If your NATPARM specifies a workfile, the import utility automatically uses the workfile.

When the data is imported or scanned, a message informs you of its successful completion. If the import operation was not successful, a message explains the problem.

## Transferring Groups

Use the Export Groups and Import Groups utilities to copy groups and their users from one Construct Spectrum Administration subsystem file to another.

### Accessing the Group Transfer Utilities

- To access the group transfer utilities:
  - 1 Enter “SA” in the Function field of the Construct Spectrum Administration Subsystem Main Menu.  
The System Administration Main Menu is displayed.
  - 2 Enter “MM” in the Function field of the System Administration Main Menu.  
The System Administration Maintenance Menu is displayed.
  - 3 Enter “DT” in the Function field of the System Administration Maintenance Menu.

The Transfer Groups Menu is displayed:

```

BS_XFER      ***** Construct Spectrum Administration Subsystem *****   CDLAYMN1
Mar 20      - Transfer Groups Menu -                                       10:30 AM

                Functions
                -----
                XG  Export Groups
                IG  Import Groups

                ?  Help
                .  Terminate
                -----
Function ..... _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit          flip                                  main

```

### Transfer Groups Menu

- 4 Enter the code in the Function field for the utility you would like to access. The window for the utility appears.

## Exporting Groups

- To export groups and users:
  - 1 Enter “XG” in the Function field of the Transfer Groups Menu. The Export Groups window appears:

```

                                Export Groups
      BS_EXGRP                                BSEXGRP0

                                Groups
                                From..... _____
                                To..... _____
      Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---
                                help retnr quit          flip

```

### Export Groups

- 2 Specify the group or range of groups to be exported. You have the following options:
  - To export one group, type it in the From field.
  - To export a range of groups, type the beginning and ending values in the From and To fields, respectively.
  - To export all groups with the same initial characters, type the characters followed by an asterisk (\*) in the From field.
  - To export all groups up to and including a specific group, type the group in the To field.
- 3 Press Enter to start the export process. If you are exporting to a PC file, the Download dialog appears, in which you can specify the file into which the data will be transferred.

When the data is exported, a message informs you of its successful completion. If the export procedure was not successful, a message explains the problem.

## Importing Groups

- To import groups:
  - 1 Log on to the library to which you wish to import the groups and invoke the Construct Spectrum Administration subsystem.
  - 2 Access the Transfer Groups Menu.
  - 3 Enter “IG” in the Function field of the Transfer Groups Menu.  
The Import Groups window appears:

```

                                Import Groups
      BS_IMGRP                                BSIMGRP0

      Import Groups.....: X
      <OR>
      Only scan input file and generate report.: _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---
      help  retrn quit          flip
```

### Import Groups

- 4 Specify the options you wish to use:
  - To import data to the current library, mark Import Groups.
  - If you wish to only create a report of the data to be imported, mark the “Only scan input file and generate report” field. The report lists the groups and users in the data file. The report is written to the default device for Report 1, for example a printer, set up by your system administrator.
- 5 Press Enter.  
If your NATPARM specifies a PC file, the Upload Data dialog appears, in which you can locate and select the file to be imported or scanned. If your NATPARM specifies a workfile, the import utility automatically uses the data from the workfile.

When the data is imported or scanned, a message informs you of its successful completion. If the import operation was not successful, a message explains the problem.

## Using the Data Transfer Utilities in Batch Mode

You can use the following JCL examples to call the data transfer utilities in batch.

### Exporting Domains

```
. . .
//CMPRT01 DD SYSOUT=X
//CMWKF01 DD DSN=DEV.DOMAINS.TEMP,DISP=(,CATLG),UNIT=SYSDA,
//          DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628),
//          SPACE=(CYL,(5,2),RLSE)
//SYSIN    DD *
GLOBALS IM=D
LOGON SYSSPEC
BS_EXDOM <from domain> <to domain>
```

### Importing Domains

```
. . .
//DDCARD DD *
ADARUN MODE=MULTI,DBID=019,PROG=USER,SVC=247
//CMPRT01 DD SYSOUT=X
//CMWKF01 DD DSN=DEV.DOMAINS.TEMP,DISP=SHR
//CMSYNIN DD *
//SYSIN    DD *
GLOBALS IM=D
LOGON SYSSPEC
BS_IMDOM X,,X << --- use this to replace steplibs
BS_IMDOM X,,, << --- use this to not replace steplibs
BS_IMDOM X,X, << --- use this to only scan the import file
```

*Note: If you mark the Replace Steplibs and Scan Import file options, the report will indicate which steplibs would have been replaced.*

## Exporting Groups

```
. . .
//CMPRT01 DD SYSOUT=X
//CMWKF01 DD DSN=DEV.DOMAINS.TEMP,DISP=(,CATLG),UNIT=SYSDA,
//          DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628),
//          SPACE=(CYL,(5,2),RLSE)
//SYSIN    DD *
GLOBALS IM=D
LOGON SYSSPEC
BS_EXGRP <from group> <to group>
```

## Importing Groups

```
. . .
//DDCARD DD *
ADARUN MODE=MULTI,DBID=019,PROG=USER,SVC=247
//CMPRT01 DD SYSOUT=X
//CMWKF01 DD DSN=DEV.GROUPS.TEMP,DISP=SHR
//CMSYNIN DD *
//SYSIN    DD *
GLOBALS IM=D
LOGON SYSSPEC
BS_IMGPR X,, << --- use this to import groups
BS_IMGPR,X, << --- use this to only scan the import file
```

## Layout of the Export Users and Groups Workfile

If you wish to import table data from external systems, for example, an in-house security system, you can create your own export utility that writes out your data into a transfer file that conforms to the import file specifications expected by the Construct Spectrum Import Groups utility. To help with this effort, Spectrum supplies module BSSI\_EX1 as an example template for a program that uses the workfile definitions described below. It is included in the SYSSPEC library.

The workfile is configured into variable length record types, with each record consisting of a line in an ASCII file. Lines are separated by CR+LF characters.

Three different types of records are supported, each of which is identified by a record-id. The information contained in each record must strictly follow the guidelines outlined in the table below with regard to data values and relative positioning within the record.

## The Header Record

This record must be the first record in the workfile. It is used to identify the version of the utility that created the file, and the version of the Spectrum system file from which the data was exported.

| <b>Field</b>    | <b>Value</b> | <b>Start<br/>Position</b> | <b>End<br/>Position</b> | <b>Length</b> | <b>Description</b>                        |
|-----------------|--------------|---------------------------|-------------------------|---------------|---|
| Record Type     | <H           | 1                         | 2                       | 2             | Identifies the start of the Header record |
| Utility Version | 4.1.2        | 3                         | 12                      | 10            | Version of the export utility             |
| Data Version    | 0412         | 13                        | 16                      | 4             | Version of the Spectrum system file       |
| Reserved        |              | 17                        | 155                     | 139           | Reserved for future use                   |
| Record Type     | H>           | 156                       | 157                     | 2             | Identifies the end of the Header record   |

## The Group Record

Group records contain the information found in the Spectrum Groups table.

| <b>Field</b> | <b>Value</b> | <b>Start<br/>Position</b> | <b>End<br/>Position</b> | <b>Length</b> | <b>Description</b>                       |
|--------------|--------------|---------------------------|-------------------------|---------------|--|
| Record Type  | <G           | 1                         | 2                       | 2             | Identifies the start of the Group record |
| Group ID     |              | 3                         | 10                      | 8             | Unique Group ID                          |
| Group Name   |              | 11                        | 60                      | 50            | Descriptive group name                   |
| Record Type  | G>           | 61                        | 62                      | 2             | Identifies the end of a Group record     |

## The User Record

User records contain the information found in the Spectrum Users table. When creating a groups and users export file, make sure that any group that is linked to a user is written to the workfile *before* the user record that refers to the group.

*Note: When you use the Groups Export utility, user records are written out to the workfile in an encrypted format. When you write your own export utility to create an export workfile, passwords cannot be encrypted because the Import Groups utility will not be able to decrypt the password before it updates the User table.*

| Field                      | Value  | Start Position | End Position | Length | Description  |
|----------------------------|--------|----------------|--------------|--------|--|
| Record Type                | <U     | 1              | 2            | 2      | Identifies the start of a User record  |
| User ID                    |        | 3              | 10           | 8      | Unique User ID   |
| Password                   |        | 11             | 50           | 40     | The user's password  |
| User Name                  |        | 51             | 100          | 50     | The user's name  |
| Debug Library              |        | 101            | 108          | 8      | The Debug Library name must be a valid Natural library name                                |
| Debug Module Derive Method | U or T | 109            | 110          | 1      | Identifies whether debug text members should be derived using a timestamp or the user's ID |
| User's Language            | 1-60   | 111            | 113          | 3      | Valid Natural *language value, zero-filled, right-justified                                |

| <b>Field</b>          | <b>Value</b> | <b>Start<br/>Position</b> | <b>End<br/>Position</b> | <b>Length</b> | <b>Description</b>  |
|-----------------------|--------------|---------------------------|-------------------------|---------------|---|
| Linked Group<br>Count | 1-10         | 114                       | 116                     | 3             | Number of groups the user is linked to, zero-filled, right-justified. This count must correspond to the number of entries in the following field  |
| Linked Group<br>IDs   |              | 117                       | 196                     | 80            | Each Group ID written to the workfile in a previous group record. Redefined into 10 occurrences of eight bytes each. Group IDs are left-justified within the eight bytes allocated to each Group ID |
| Record Type           | U>           | 197                       | 198                     | 2             | Identifies the end of a User record   |

---

## APPENDIX A — USING BSSPARMN TO PARSE USER PARAMETERS

The BSSPARMN routine is supplied with Construct Spectrum to parse the keywords you specify in the Service Start parameters fields in the Maintain Services panel. For more information, see **Defining Service Start Parameters and Service Start Routines**, page 59.

BSSPARMN uses the parameter data areas BSSPARMA and CDPDA-M. These data areas define the interface to this routine and are described in the following sections.

### BSSPARMA

This data area defines the parameter string as well as the valid keywords. It returns the individual parameters after they have been parsed. The following fields are defined in the data area:

| <b>Input Parameter</b> | <b>Description</b>   |
|------------------------|--|
| PARM (A1/1:500)        | This string defines the parameter values that were entered in the <u>User</u> parameters fields. On the input screen, this string is broken up into five blocks of 50 characters. Normally, this parameter contains a series of keyword=value combinations separated by commas (or alternate Natural input delimiter characters). If the value contains special characters, it must be placed within quotes. |
| C#KEYWORD (I2)         | This variable tells BSSPARMN how many valid keywords are supplied in the KEYWORD field.  |

| <b>Input Parameter</b> | <b>Description (continued)</b>  |
|------------------------|---|
| KEYWORD (A32/1:15)     | For every valid keyword that can be entered, specify an entry in this array. Also assign C#KEYWORD to indicate the number of valid keywords passed.             |
| SHOW-RESULTS (L)       | If you wish to supply a test or trace facility, this flag can be set to have BSSPARMN write out the individual keyword values to the screen after parsing them. |

| <b>Output Parameter</b>  | <b>Description</b>  |
|--------------------------|---|
| C#INDIVIDUAL-PARMS (I2)  | This value indicates the number of individual parameter values that were identified by BSSPARMN.  |
| KEYWORD-ENTERED (L/1:15) | This field is set to true for occurrence <i>n</i> , if the <i>n</i> th keyword in the KEYWORD array was entered followed by an input assign character. Note that the input assign character is normally an equal (=) sign, but it can be overridden by the IA Natural Parameter.                            |
| VALUE (A200)             | This field contains the value of a parameter. If the corresponding KEYWORD-ENTERED flag is set, this value corresponds to the KEYWORD value in the same occurrence. If KEYWORD-ENTERED is false, there may still be a value in the corresponding VALUE field, but this value was not preceded by a keyword. |

## CDPDA-M

If BSSPARMN was not able to parse the input parameters, CDPDA-M contains a message number in CDPDA-M.##MSG-NR to indicate the reason. This message number corresponds to a SYSERR message for application SYSSPEC. Substitution parameters may also be returned in CDPDA-M.##MSG-DATA(1).

*Example of using BSSPARMN*

```

DEFINE DATA
  LOCAL USING BSSPARMA
  LOCAL USING CDPDA-M
  LOCAL
    01 #I(I1)
END-DEFINE
*
* Parameter values in PARM-CHUNK are normally entered as INPUT
* values
BSSPARMA.PARM-CHUNK(1) :=
  'JOB-NAME=SOMEJOB,NATPARAM="FNAT=(1,2),PROFILE=SYSSPEC"'
*
* define 3 valid parameters
BSSPARMA.C#KEYWORD := 3
BSSPARMA.KEYWORD(1) := 'NATPARAM'
BSSPARMA.KEYWORD(2) := 'USER'
BSSPARMA.KEYWORD(3) := 'JOB-NAME'
*
* Call BSSPARMN to parse the parameters
CALLNAT 'BSSPARMN' BSSPARMA CDPDA-M
*
* If there were no errors, show the results
IF CDPDA-M.##RETURN-CODE = ' ' THEN
  FOR #I = 1 TO BSSPARMA.C#INDIVIDUAL-PARMS
    DECIDE FOR FIRST CONDITION
      WHEN BSSPARMA.KEYWORD-ENTERED(#I)
        PRINT BSSPARMA.KEYWORD(#I) 'equals'
          BSSPARMA.VALUE(#I) (AL=50)
      WHEN BSSPARMA.VALUE(#I) NE ' '
        PRINT 'Value without keyword' BSSPARMA.VALUE(#I) (AL=50)
      WHEN NONE
        IGNORE
    END-DECIDE
  IF NOT BSSPARMA.KEYWORD-ENTERED(#I) AND #I LE BSSPARMA.C#KEYWORD
  THEN
    PRINT 'No value entered for parameter' BSSPARMA.KEYWORD(#I)
  END-IF
  END-FOR
ELSE
  WRITE 'Parameter error' CDPDA-M.##MSG-NR CDPDA-M.##MSG-DATA(1)
END-IF
END

```



---

## APPENDIX B — USING SPSERRN FOR ADDITIONAL ERROR-HANDLING

The SPSERRN user exit is supplied with Construct Spectrum for enhancing the core error-processing. The user exit allows you to add special error-processing such as sending notification of errors to the operator.

SPSERRN provides the following information:

- Error number
- Line number
- Status
- Program name
- Program level

SPSERRN uses the parameter data area CDERRLDA.

*Note: Do not STOP or TERMINATE processing in this subprogram or any associated modules. Control should return to the Spectrum error handler to ensure complete tracking of internal Spectrum information.*

*Note: If you use SPSERRN to code additional error-processing, be sure to back up the program before upgrading Construct Spectrum in the future. Then move the program back after the installation is complete.*



# INDEX

\*TIMESTMP system variable, 151

## A

ACF2 security, 45

ActiveX business object

- role in architecture of Construct applications, 30

Administration subsystem

- See Construct Spectrum

- Administration subsystem

Application administration, 33

Application service definition

- and Spectrum security, 124

- role in application administration, 34

Architecture of Construct Spectrum

applications, 23

ASCII character set

- translation of, 25, 27

Attach server

- activation policy, 58

- and SUBTASKB service start

- routine, 64

- in Spectrum Service Manager, 148

- launching services in batch or as subtasks, 58

- linking services to, 57

- runtime error-handling, 58

Attach service, 48

- and the START command, 59

- defining, 51

- name supplied with Construct Spectrum, 57

- specifying name, 56

Attribute file, Entire Broker, 70

## B

Batch

- initiating services using, 59

BATCHJOB service start routine, 61

- and JCL keywords, 61

- example of, 63

- NATRJE, 60

Broker Driver program

- Automatic logon, 156

- Conversation ID, 156

- function of, 143

- Security token, 156

- Service, 156

- using, 154

Broker ID, 55

BS\_SYNCN exit module, 136

BS\_USEC\* security exits, 133

BS\_USEC1 security exit, 134

BS\_USEC2 security exit, 135

BSSI\_EX1 module, 175

BSSPARMA parameter data area, 179

BSSPARMN routine, 179

## C

- Callnat service, 49
- CALLNAT simulation, 28
- CA-TOP SECRET security, 45
- CDERRLDA parameter data area, 183
- CDPDA-M parameter data area, 180
- Change Password program
  - function of, 143
  - using, 149
- Character UI
  - role in architecture of Construct applications, 24
- Cleanup queue, 141
- Client service, 48
  - defining, 51
  - displaying in Manage Services panel, 73
- Client tools, 143
- Command service, 49
  - CALLNAT, 49
  - ENVIRONMENT, 49
  - ERROR, 49
  - PING, 49
  - SHUTDOWN, 49
  - STEPLIBS, 50
  - syntax, 50
  - SYSPROF, 50
  - TRACE, 49
- Common server startup interface, 58
- Common start facility, 48
- Communication logs
  - viewing from Manage Services panel, 75
- Compression/decompression of data, 25, 27
- Construct Spectrum
  - documentation, 20
  - runtime environment, 158
  - security cache, 35
- Construct Spectrum Administration subsystem
  - application administration, 33
  - deploying, 158
  - invoking, 36
  - uses, 32
  - using EntireX Security, 45
  - using menus, 37
  - using menus with Natural Security, 43
  - using Natural Security, 40
- Construct Spectrum SDK
  - documentation, 19
- Construct Spectrum service
  - attach service, 48, 51
    - See also Attach service
  - client service, 48, 51
    - See also Client service
  - command service, 49
    - See also Command service
  - debugging, 77
  - definition of, 47
  - initiating, 76, 78
  - main service, 49
    - See also Main service
  - names supplied with Construct Spectrum, 57
  - pinging, 76, 80
  - refreshing, 76, 80
  - server service, 48, 51
    - See also Server service
  - shutting down, 76, 81
- Control record
  - accessing log, 94
  - enabling Natural Security, 92
  - Entire Net-Work timeout, 93
  - modifying control data, 92
  - security exit, 92
  - uses of, 89
- Conventions, typographical, 17

Conversation factory  
in Spectrum Service Manager, 148

## D

Data compression/decompression, 25, 27

Data encryption/decryption, 25

Data transfer utilities, 165

export

creating, 175

Export Domains, 165, 167

export domains in batch, 173

Export Groups, 171

export groups in batch, 174

Export Users and Groups

Workfile, 175

Import Domains, 165, 168

import domains in batch, 173

Import Groups, 172

import groups in batch, 174

using in batch mode, 173

Data translation, 25, 27

Debug command, 76–77

Debug library and file, 105

Debugging information, 26

Decryption/encryption of data, 26

Deploying

Construct Spectrum Administration  
subsystem

attach service, 163

copying system files, 162

data transfer utilities, 165

dispatch service, 163

example of, 159

Natural Security, 163

prerequisite software, 161

security service, 163

Spectrum system file database, 162

transferring domains, 164  
transferring group and user  
tables, 164

Construct Spectrum application, 158

Description text box

in Spectrum Service Manager, 147

Disabled security status, 132

Dispatch service

encapsulation of Entire Broker  
calls, 28

in program parameters, 69

in Spectrum Service Manager, 147

name supplied with Construct  
Spectrum, 57

role in architecture of Construct  
applications, 25

security mode, 69

system functions, 26

Documentation, related, 19

Construct Spectrum, 20

Construct Spectrum Software

Development Kit, 19

Natural Construct, 20

Domain

exporting, 167

exporting in batch, 173

importing, 168

importing in batch, 173

role in application administration, 34

role in deploying, 164

role in Spectrum security, 96

transfer utilities

accessing, 166

transferring, 165

## E

EBCDIC character set  
translation of, 25, 27

- Enabled security status, 132
- Encryption/decryption of data, 25–26
- EndJob command, 76–77
- EndService command, 76, 78
- Entire Broker
  - attribute file
    - changing, 70
  - encapsulation of calls by SDC, 28
  - parameters
    - in Spectrum Service Manager, 147
  - role in architecture of Construct applications, 26–27, 30
- Entire Broker service settings, 52, 54
  - attach service name, 56
  - broker ID, 55
  - server class, 55
  - server name, 56
  - service, 56
  - user ID and password, 56
- Entire Net-Work timeout, 93
- EntireX security
  - using with Construct Spectrum, 45
- EnVironment command, 76, 78
- Environment service, 49
- Error service, 49
- Error-handling, 27
  - and attach server, 58
  - CDERRLDA PDA, 183
  - SPSERRN user exit, 183
- ETB02150148 error, 145
- Export Domains utility, 165
- Export Users and Groups Workfile, 175
  - group record, 176
  - header record, 176
  - user record, 177

## G

- Granted security status, 132
- Group
  - associating with user
    - in Natural Security, 109
    - in Spectrum security, 106
  - defining
    - using Natural Security, 100
    - using Spectrum security, 99
  - exporting, 171
  - exporting in batch, 174
  - importing, 172
  - importing in batch, 174
  - role in Spectrum Security, 34
  - role in Spectrum security, 96
  - transferring, 164, 169
- Group transfer utilities
  - accessing, 169
- GUI dialog
  - role in architecture of Construct applications, 29

## I

- IIS
  - role in architecture of Construct applications, 30
- Import Domains utility, 165
- Information logs, 32
- Initiate command, 76, 78
- Internet/intranet
  - role in architecture of Construct applications, 31

**J**

JCL keyword  
 and BATCHJOB service start routine, 61

JOB-NAME JCL keyword  
 and BATCHJOB service start routine, 62

JOB-PREFIX keyword  
 and BATCHJOB service start routine, 61

JOB-TRANSACTION JCL keyword  
 and BATCHJOB service start routine, 62

**L**

Library image file  
 simulated PDAs in, 28

LIBRARY JCL keyword  
 and BATCHJOB service start routine, 62

Library Maintenance / Link Users window  
 accessing, 111  
 using, 112

LIF, see Library image file

Logical File 135, 162

Logical File 136, 162

Logs, information, 32

**M**

Main service, 49

Maintain Control Record window  
 accessing, 90

enabling Natural Security, 92

Entire Net-Work timeout, 93

modifying control data, 92

security exit, 92

Maintain Group Profile window  
 accessing, 101  
 using, 102

Maintain Group Table panel  
 accessing, 99

Maintain Object Security by Group panel, 128  
 Clear action, 128

Maintain Object Security panels  
 accessing, 125

Maintain Services panel 1  
 attach, 57  
 Description field, 54  
 Entire Broker service settings, 52, 54  
 attach service name, 56  
 broker ID, 55  
 server class, 55  
 server name, 56  
 service, 56  
 user ID and password, 56  
 Name field, 54  
 service start parameters, 52, 59  
 service start routine, 59–60

Maintain Services panel 2, 67  
 default language, 68  
 program parameters, 69  
 attach client, 70  
 dispatch service, 69  
 Natural program name, 70  
 service timeout, 68  
 service type, 68  
 unique ID, 69

Maintain Services panels  
 accessing, 51

Maintain User Profile window  
 accessing, 107  
 using, 108

Maintain User Table panel  
  accessing, 104  
  debug filename, 106  
  debug library, 105  
  groups, 106

Maintain User's Groups window  
  accessing, 110  
  using, 111

Manage Services panel  
  accessing, 73  
  Debug command, 76  
  displaying client services, 73  
  EndJob command, 76  
  EndService command, 76  
  EnVironment command, 76  
  Initiate command, 76  
  Last command field, 75  
  Ping command, 76  
  purpose of, 72  
  Refresh command, 76  
  Repeat command field, 75  
  Replica ID field, 74  
  Respondent field, 75  
  Service field, 74  
  Service type field, 74  
  SHutdown command, 76  
  STeplibs command, 76  
  SYsprof command, 76  
  viewing Communication logs, 75

Message-handling, 27

Method  
  role in application administration, 34  
  role in Spectrum security, 96

Microsoft Internet Information Server  
  role in architecture of Construct  
  applications, 30

## N

Name text box  
  in Spectrum Service Manager, 146

NATPARM JCL keyword, 64  
  and BATCHJOB service start  
  routine, 62

Natural profile  
  adding, 71

Natural Security  
  accessing query panels, 117  
  and SUBTASKB service start  
  routines, 65  
  configuring for Construct  
  Spectrum, 40  
  enabling in the control record, 92  
  interface to restricted libraries, 41  
  Library Maintenance / Link Users  
  window, 111–112  
  Maintain Group Profile window, 101  
  Maintain User Profile window, 107–  
  108  
  Maintain User's Groups window, 110–  
  111  
  Query Group Table window, 118  
  Query User Table window, 119  
  Query User's Groups window, 120  
  role in deploying Construct Spectrum  
  Administration subsystem, 163  
  using with Construct Spectrum, 40

Natural subprogram  
  role in architecture of Construct  
  applications, 24

No Privileges security status, 131

NSCUSEX1 module, 41

## O

- Object
  - role in Spectrum security, 96
- Optimistic security checking, 135

## P

- Parameter data areas
  - simulation by Spectrum Dispatch Client, 28
- PASSWORD JCL keyword
  - and BATCHJOB service start routine, 63
- Pessimistic security checking, 134
- Ping command, 76, 80
- Ping service, 49
- PROGPARM JCL keyword
  - and BATCHJOB service start routine, 63
- Program parameters for services, 69
  - attach client, 70
  - dispatch service, 69

## Q

- Query Communications Logs panel
  - accessing, 83
- Query Log panels
  - accessing, 87
- Query panels
  - accessing, 85

## R

- RACF security, 45
- Refresh command, 76, 80
- Remote Clock program
  - command line parameters, 152
  - function of, 143
  - using, 151
  - using Spectrum Dispatch Client, 152
- Resetting the Spectrum security cache, 45
- Response time-out
  - in Spectrum Service Manager, 148
- Revoked security status, 132
- Runtime environment, 158

## S

- Security administration, 34
- Security cache, 35
  - synchronizing using BS\_SYNCN, 136
  - synchronizing with cleanup queue, 141
- Security exit, 92, 123–124
  - BS\_USEC1, 134
  - BS\_USEC2, 135
  - BS\_USECO, 133
  - using security exit modules, 133
- Security for Group-Domain window, 129
- Security for Object window, 130
  - Clear action, 131
- Security mode
  - for dispatch service, 69
  - in Spectrum Service Manager, 148
- Security service
  - name supplied with Construct Spectrum, 57

- role in architecture of Construct applications, 26
- Security settings
  - adding group-domain combination, 126
  - defining, 125
  - options for group-domain combination, 127
  - options for methods, 130
  - options for objects, 128
  - purging group-domain combination, 127
  - statuses
    - Disabled, 132
    - Enabled, 132
    - Granted, 132
    - No Privileges, 131
    - Revoked, 132
- Server class, 55
- Server name, 56
- Server service, 48
  - defining, 51
- Server startup process
  - attach server, 57
  - automating with main and command services, 49
  - initiating services as subtasks, 59
  - initiating services in batch, 59
  - initiating services online, 59
  - START command, 58
- Service
  - in Entire Broker service settings, 56
- Service ID
  - in Spectrum Service Manager, 146
- Service Properties dialog, 145
- Service Start parameters, 52
  - BSSPARMA PDA, 179
  - CDPDA-M PDA, 180
  - parsing with BSSPARMN, 179
- Service start routine
  - BATCHJOB, 59–61
  - generating trace output, 65
  - SUBTASK, 59–60
  - SUBTASKB, 64
    - STACK parameter, 64
  - using, 60
  - writing to print and workfile, 65
- Service startup process
  - service start parameters, 59
  - service start routine, 59–60
  - under Natural Security, 65
- Service timeout, 68
- Service type
  - in Spectrum Service Manager, 147
- SHutdown command, 76, 81
- Shutdown service, 49
- Shutting down services
  - all replicas, 81
  - all servers, 81
  - single replica, 81
- Spectrum administration
  - role in architecture of Construct applications, 25
- Spectrum Control Record Log
  - accessing, 94
- Spectrum Dispatch Client
  - CALLNAT simulation, 28
  - database transaction control by, 29
  - encapsulation of Entire Broker calls, 28
  - role in architecture of Construct applications, 27, 30
- Spectrum dispatch service
  - in program parameters, 69
  - in Spectrum Service Manager, 147
  - role in architecture of Construct applications, 25
  - role in deploying Construct Spectrum Administration subsystem, 163

- system functions, 26
- Spectrum security
  - accessing queries and logs, 114, 137
  - accessing Query Log panels, 116
  - and version of application service definition, 124
  - disabled status, 124
  - enabled status, 124
  - for applications, 123
  - levels of security, 124
- Spectrum security service, 158
  - role in architecture of Construct applications, 26
  - role in deploying Construct Spectrum Administration subsystem, 159, 163
- Spectrum service
  - attach service, 48, 51
    - See also Attach service
  - client service, 48, 51
    - See also Client service
  - command service, 49
    - See also Command service
  - copying definitions, 149
  - debugging, 77
  - definition, 32
  - initiating, 76, 78
  - main service, 49
    - See also Main service
  - pinging, 76, 80
  - pinging using the Spectrum Service Manager, 143
  - refreshing, 76, 80
  - server service, 48
    - See also Server service
  - shutting down, 76, 81
  - starting in batch mode under Natural Security, 41
- Spectrum Service Manager
  - function of, 143
  - Service Properties dialog, 145
    - Description, 147
    - Entire Broker Parameters, 147
    - Name, 146
    - Response time-out, 148
    - Security mode, 148
    - Service ID, 146
    - Service type, 147
- SPSEERN user exit, 183
- SPSSTART common start facility, 48, 59
- START command, 58
  - specifying Natural program name, 70
  - to initiate services in batch, 59
- Starting services, 52
- Steplib
  - role in application administration, 33
- STeplibs command, 76, 83
- Steplibs service, 50
- Subprogram proxy
  - and Spectrum dispatch service, 25
  - role in architecture of Construct applications, 24
- Subtask
  - initiating services as, 59
  - of attach server, 58
- SUBTASK service start routine
  - CMTASK, 60
- SUBTASKB service start routine, 64
  - example of, 64
  - NATPARM keyword, 64
  - STACK parameter, 64
  - TRANSACTION keyword, 64
- Synchronizing the security cache, 136
- Synchronizing the security cache with the cleanup queue, 141
- SYSprof command, 76, 83
- Sysprof service, 50
- System administration tools
  - accessing, 140

## T

- Timeout
  - for service, 68
- Timestamp service
  - and Remote Clock program, 151
  - name supplied with Construct Spectrum, 57
- Tools
  - client, 143
  - system administration
    - accessing, 140
- TRACE command, 65
  - example of, 66
- Trace output
  - generating, 65
- Trace service, 49
- TRANSLATION Entire Broker attribute, 70
- Translation of character sets, 25, 27
- Typographical conventions, 17

## U

- User
  - associating with groups
    - in Natural Security, 109
    - in Spectrum security, 106
  - defining, 104
    - using Natural Security, 107
    - using Spectrum security, 104
  - defining debug library and file, 105
  - role in Spectrum security, 96
  - transferring, 164
- User ID
  - role in security administration, 34
- User ID and password
  - in Entire Broker service settings, 56

- USER-ID JCL keyword
  - and BATCHJOB service start routine, 62

## V

- Visual Basic business object
  - role in architecture of Construct applications, 29

## W

- Web application
  - role in architecture of Construct applications, 30