GENESYS®
AN ALCATEL·LUCENT COMPANY

Statistics SDK 7.6

Web Services

API Reference

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the *Genesys 7 Licensing Guide*.

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76sdk_ref_stat-ws_11-2008_v7.6.101.00

Table of Contents

Statistics SDK 7.6 Web Services

Preface

Welcome to the *Statistics SDK 7.6 Web Services API Reference*. This reference documents how the Genesys Integration Server (GIS) SOAP Protocol interface exports information retrieved from Genesys Stat Server through the Statistics application programming interface (API). In particular, this reference introduces you to the elements of the Web Services Description Language (WSDL) files for the Session Service and the Statistics SDK (software development kit) Service.

This document is valid only for the 7.6 release(s) of this product.

Note: For releases of this document created for other releases of this product, please visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com`.

This chapter provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information. It contains the following sections:

The Statistics SDK Service provides access to statistical information gathered by Stat Server. Using the Statistics SDK Service you can:

- Log in, log out, and verify licensing that enables your client application to use the Statistics SDK Service.

- Use the Statistics SDK Service to request statistical information from Stat Server.

- Retrieve and view certain configuration information from the Genesys Configuration Database.

Intended Audience

This reference, primarily intended for developers who are familiar with Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML) technologies, assumes that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.
- Network design and operation.
- Your own network configurations.

You should also be familiar with the following technologies:

- XML Schemas
- SOAP Protocol
- WSDL
- XPath

Depending on the technology choice for client development, working knowledge of Java or another programming language may be required.

Developers should also be familiar with the Genesys Framework, especially Stat Server.

Usage Guidelines

The Genesys developer materials outlined in this document are intended to be used for the following purposes:

- Creation of contact-center agent desktop applications associated with Genesys software implementations.
- Server-side integration between Genesys software and third-party software.
- Creation of a specialized client application specific to customer needs.

The Genesys software functions available for development are clearly documented. No undocumented functionality is to be utilized without Genesys's express written consent.

The following Use Conditions apply in all cases for developers employing the Genesys developer materials outlined in this document:

1. Possession of interface documentation does not imply a right to use by a third party. Genesys conditions for use, as outlined below or in the *Genesys Developer Program Guide,* must be met.

2. This interface shall not be used unless the developer is a member in good standing of the Genesys Interacts program or has a valid Master Software License and Services Agreement with Genesys.

3. A developer shall not be entitled to use any licenses granted hereunder unless the developer's organization has met or obtained all prerequisite licensing and software as set out by Genesys.

4. A developer shall not be entitled to use any licenses granted hereunder if the developer's organization is delinquent in any payments or amounts owed to Genesys.

5. A developer shall not use the Genesys developer materials outlined in this document for any general application development purposes that are not associated with the above-mentioned intended purposes for the use of the Genesys developer materials outlined in this document.

6. A developer shall disclose the developer materials outlined in this document only to those employees who have a direct need to create, debug, and/or test one or more participant-specific objects and/or software files that access, communicate, or interoperate with the Genesys API.

7. The developed works and Genesys software running in conjunction with one another (hereinafter referred to together as the "integrated solutions") should not compromise data integrity. For example, if both the Genesys software and the integrated solutions can modify the same data, then modifications by either product must not circumvent the other product's data integrity rules. In addition, the integration should not cause duplicate copies of data to exist in both participant and Genesys databases, unless it can be assured that data modifications propagate all copies within the time required by typical users.

8. The integrated solutions shall not compromise data or application security, access, or visibility restrictions that are enforced by either the Genesys software or the developed works.

9. The integrated solutions shall conform to design and implementation guidelines and restrictions described in the *Genesys Developer Program Guide* and Genesys software documentation. For example:

   a. The integration must use only published interfaces to access Genesys data.

   b. The integration shall not modify data in Genesys database tables directly using SQL.

   c. The integration shall not introduce database triggers or stored procedures that operate on Genesys database tables.

Any schema extension to Genesys database tables must be carried out using Genesys Developer software through documented methods and features.

The Genesys developer materials outlined in this document are not intended to be used for the creation of any product with functionality comparable to any Genesys products, including products similar or substantially similar to Genesys's current general-availability, beta, and announced products.

Any attempt to use the Genesys developer materials outlined in this document or any Genesys Developer software contrary to this clause shall be deemed a

material breach with immediate termination of this addendum, and Genesys shall be entitled to seek to protect its interests, including but not limited to, preliminary and permanent injunctive relief, as well as money damages.

Chapter Summaries

In addition to this opening chapter, this document contains the following chapters:

- Chapter 1, "Overview," on page 17 explains how GIS (Genesys Integration Server) interacts with the Genesys Framework, on one hand, and with your client application via the Session Service and the Statistics SDK Service, on the other.

- Chapter 2, "Using the Session Service," on page 21 presents a complete description of the Session service namespaces, messages, operations, and so on.

- Chapter 3, "View Configuration Methods," on page 33 details the methods available to access read-only configuration information for users who do not have a Configuration SDK Service license.

- Chapter 4, "Statistics SDK Service Namespaces," on page 37 introduces the Statistics SDK Service and lists the namespaces used by the Statistics WSDL file.

- Chapter 5, "See also "timeIntervalType" on page 52, and the examples of this complex type being used in a request in Figure 9 on page 81.," on page 69 presents the data types that define the structure of the information contained in a message. The WSDL file uses standard XML schema types and Genesys types.

- Chapter 6, "Statistics SDK Service Messages," on page 71 introduces the `request`/`response` schema that transmits information between a client application and the GIS.

- Chapter 7, "Statistics SDK Service Operations," on page 83 defines the Statistics WSDL operations. These operations taken together are considered the Statistics SDK Service.

- Chapter 8, "Statistics SDK Service Bindings," on page 97 explains how bindings attach a specific protocol (SOAP/HTTP) to each WSDL operation.

Document Conventions

This document uses some stylistic and typographical conventions with which you might want to familiarize yourself.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

`76fr_ref_02-2008_v7.6.001.00`

You will need this number when you are talking with Genesys Technical Support about this product.

Type Styles

Italic

In this document, italic is used for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples: • Please consult the *Genesys 7 Migration Guide* for more information.

• *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.

• Do *not* use this value for this option.

• The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

Examples: • Select the `Show variables on screen` check box.

• Click the `Summation` button.

• In the `Properties` dialog box, enter the value for the host server in your environment.

• In the `Operand` text box, enter your formula.

• Click `OK` to exit the `Properties` dialog box.

- The following table presents the complete set of error messages T-Server®️ distributes in `EventError` events.

- If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example: •   Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

- The *Statistics SDK 7.6 Web Services Developer's Guide,* which explains how to set up the Statistics SDK code examples and presents the text of the examples along with explanatory comments.

- The *Genesys Integration Server 7.6 Deployment Guide,* which provides installation, configuration, and starting and stopping instructions for GIS.

- The *Configuration SDK 7.6 Web Services Developer's Guide,* which explains how to set up the Configuration SDK code examples, including those that provide read-only configuration information for users with a Statistics SDK Web Services license only, and presents the text of the examples along with explanatory comments.

- The *Configuration SDK 7.6 Web Services API Reference,* which details the messages, operations, data types, and constants as defined in the Configuration WSDL file that governs communication between the client application and GIS.

- The *Framework 7.x Solution Control Interface Help*, which provides specific instructions on using Solution Control Interface to start, stop, and monitor Genesys solutions.

- The *Framework 7.x Stat Server User's Guide,* which contains instructions on using Stat Server. See especially the "Custom Value Statistic Types" section.

- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.

- The *Genesys Migration Guide*, also on the Genesys Documentation Library DVD, which provides a documented migration strategy from Genesys product releases 5.1 and later to all Genesys 7.x releases. Contact Genesys Technical Support for additional information.

- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

Information on supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Systems and Databases*
- *Genesys Supported Media Interfaces*

Genesys product documentation is available on the:

- Genesys Technical Support website at `http://genesyslab.com/support`.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com`.

Making Comments on This Document

If you especially like or dislike anything about this document, please feel free to e-mail your comments to `Techpubs.webadmin@genesyslab.com`.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the information in this document only and to the

way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

1 Overview

The *Statistics SDK 7.6 Web Services API Reference* documents the methods available to:

- Log in to GIS and authenticate your statistics license using the Session Service.
- Access and edit Genesys statistics using the Statistics SDK Service.
- View configuration information from the Configuration Database using a subset of Configuration SDK Service methods.

Note: Configuration information is read-only unless you have a Configuration SDK Service license as well as a Statistics SDK Service license. Also, because GIS simply passes on Configuration SDK Service messages to the Configuration Server SOAP interface, the Configuration API functions differently from the Statistics API. For details, see the *Genesys Integration Server 7.6 Deployment Guide.*

This chapter includes the following sections:

Note: This reference describes the contents of the Session and Statistics WSDL files and a subset of the Configuration WSDL methods. The third-party software and the objects and classes needed to create client applications are not discussed here. For more information on these topics, see the *Statistics SDK 7.6 Web Services Developer's Guide.*

Introduction

The operations that you can use to access Genesys Stat Server statistics are described in the WSDL file that GIS generates at your request. Each operation consists of a pair of SOAP over HTTP request and response messages passed between the client application and GIS.

These are the available operations:
- `retrieveStatistic`
- `subscribeStatistic`
- `unsubscribeStatistic`
- `retrieveSubscribedStatistics`
- `retrieveStatisticalProfile`

The format for constructing SOAP messages to perform these functions is described in the remaining chapters of this book.

Note: This document is intended as a reference. For how-to information and examples of code invoking these methods, see the *Statistics SDK 7.6 Web Services Developer's Guide*.

Generate the WSDL Files

To generate the WSDL files after deploying GIS:

1. Open any web browser.

2. Enter the appropriate URL. Note that the Session and Statistics Services each offer a choice of either SOAP RPC encoding or SOAP Document Literal encoding:

   • **Session Service (RPC encoding):**
     `http://<GIS_HOST>:<GIS_PORT>/gis/services/SessionService?wsdl`

   • **Session Service (Document Literal encoding):**
     `http://<GIS_HOST>:<GIS_PORT>/gis/services/`
     `SessionService_76_DOCUMENT?wsdl`

   • **Statistics Service (RPC encoding):**
     `http://<GIS_HOST>:<GIS_PORT>/gis/services/StatService?wsdl`

   • **Statistics Service (Document Literal encoding):**
     `http://<GIS_HOST>:<GIS_PORT>/gis/services/`
     `StatService_76_DOCUMENT?wsdl`

   • **Notification Service:**
     `http://<GIS_HOST>:<GIS_PORT>/gis/wsdl/notification.wsdl`

   • **Configuration Service:**
     `http://<GIS_HOST>:<GIS_PORT>/gis/services/CSProxyService?wsdl`

Supported Framework Versions

The Genesys Statistics SDK Service is compatible with single-tenant or multi-tenant Framework versions 6.5 and 7.*x*.

To retrieve configuration information, you must use the Configuration Server or CS Proxy SOAP interface, as documented in Chapter 3. CSProxy is available in Framework 6.5 and 7.*x*. The Configuration Server SOAP interface is available in version 7.0 and higher.

2 Using the Session Service

This chapter provides a complete reference to the methods used to log in, validate and manage licenses, and log out at the end of a session. This chapter includes the following sections:

Overview

The Session Service controls session start, session end, and licensing for all GIS-exposed SDK Services. This service is a standard component of GIS, and you do not need a license to access it.

The Session ID

If your authorization request is accepted, then the Session `loginResponse` message that is returned includes a Session ID that must be included in every other message sent during this session. Without a valid Session ID, you cannot access functions for any licensed, GIS-exposed SDK Service.

Using the Session Service

The rest of this chapter details the information available in the WSDL file. You should use this information as a reference rather than a how-to guide. For

assistance in constructing Session Service requests, including examples of request and response messages, see the *Statistics SDK 7.6 Web Services Developer's Guide*.

Generating the Session WSDL File

To generate the Session WSDL file:

1. Open a web browser.

2. Enter one of the following URLs, depending on your choice of SOAP encoding:

   - **RPC encoding:**
     `http://<GIS_HOST>:<GIS_PORT>/gis/services/SessionService?wsdl`

   - **Document Literal encoding:**
     `http://<GIS_HOST>:<GIS_PORT>/gis/services/`
     `SessionService_76_DOCUMENT?wsdl`

Session Methods and Types

The Session service uses the following methods:

- **login**—Logs the user in to GIS. The `LoginResponse` message contains the Session ID, which you must use for all subsequent messages that belong to this transaction.

- **getService**—Subscribes to a license for a specified, GIS-exposed Genesys SDK Service.

- **browseService**—Lists all licenses available to your client application.

- **releaseService**—Unlocks the license used by your application so that another user can subscribe to it.

- **logout**—Closes your transaction and logs you out of GIS.

You must specify the Session ID for all Session methods with the exception of the `login` method. None of the Session methods require a license.

Types

Within the schema are multiple data types that contain definitions for the various types used in the messages. There are three kinds of data defined in the schema:

- `element`—An element represents data. An element is associated with a data type that may be a standard, simple, or a complex data type.

- `simple type`—A simple type is always restricted. Most simple types are used as a set of enumerations that specify types of objects.

- **`complex type`**—A complex type contains one or more elements, possibly with one or more simple types or even other complex types nested within it. A complex type generally is made up of a sequence of the same type of elements, similar to an array.

Using Types in Messages

Message definitions specify type attributes for their parts. For example, the `loginRequest` message has the following part:

```
<part name="identity" type="tns1:Identity"/>
```

The name of this part is `identity`, and it requires the `tns1:Identity` type that is defined in the types section. If you check the types section, you will find that `tns1:Identity` is a complex type with a sequence of three elements: `principal`, `credentials,` and `tenant` (all of which are standard `xsd:string` elements).

This shows that the `loginRequest` message sent to GIS is a SOAP packet that contains a tenant name, user name, and password. See the code examples in the *Statistics SDK 7.6 Web Services Developer's Guide* for an example of creating and using the `loginRequest` message.

Namespaces

Namespaces are virtual domains that specify how the elements of a message should be interpreted. Namespaces allow you to give different meanings to a single element or attribute tag by adding a namespace prefix. The parser then uses this prefix to locate the specification required to interpret the tag correctly.

The Session Service uses two types of namespaces: standard, and Genesys-specific.

Standard Namespaces

Standard namespaces are public namespaces that contain specifications written and maintained in accordance with the W3 Consortium standards. The Session Service uses these standard namespaces:

- `xmlns:apachesoap="http://xml.apache.org/xml-soap"`
- `xmlns:soapenc"=http://schemas.xmlsoap.org/soap/encoding/"`
- `xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"`
- `xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"`
- `xmlns:xsd="http://www.w3.org/2001/XMLSchema"`

Genesys-Specific Namespaces

Genesys-specific namespaces define elements used only by the Genesys
Session Service:

- `targetNamespace="http://www.genesyslab.com/services/sessionservice/`
  `wsdl/2002/03"`
- `xmlns:impl="http://www.genesyslab.com/services/sessionservice/wsdl/`
  `2002/03"`
- `xmlns:intf="http://www.genesyslab.com/services/sessionservice/wsdl/`
  `2002/03"`
- `xmlns:tns1="http://www.genesyslab.com/services/sessionservice/`
  `types/2002/03"`

Types

The Session service schema includes the `Identity` and `ServiceList` complex
types, and does not use any simple types.

Identity Type

The `Identity` complex type is structured as shown in Table 1.

Table 1: Session Service Identity Type

Complex Type Name	Elements	Nillable	Type
`Identity`	`principal`—Username, as declared in Configuration Manager.	true	string
	`credentials`—Password, as declared in Configuration Manager.	true	string
	`tenant`—Tenant resource that the specified user belongs to.	true	string

ServiceList Type

The `ServiceList` complex type is a SOAP array, all of the members of which
are the `xsd:string` type.

Valid values are the names of the various GIS services.

Messages

The Session Service has six different operations, each of which uses a request and response message. Syntax for the twelve available messages is described in this section.

login Messages

Request and response messages for the `login` operation. The `LoginResponse` message includes the Session ID required for all other messages.

Syntax

```
<wsdl:message name="loginRequest">
   <wsdl:part name="identity" type="tns1:Identity"/>
</wsdl:message>

<wsdl:message name="loginResponse">
   <wsdl:part name="loginResult" type="xsd:string"/>
</wsdl:message>
```

logout Messages

Request and response messages for the `logout` operation.

Syntax

```
<wsdl:message name="logoutRequest">
   <wsdl:part name="sessionID" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="logoutResponse"></wsdl:message>
```

getService Messages

Request and response messages for the `getService` operation.

Syntax

```
<wsdl:message name="getServicesRequest">
   <wsdl:part name="services" type="tns1:ServiceList"/>
</wsdl:message>

<wsdl:message name="getServicesResponse">
   <wsdl:part name="getServicesResult" type="tns1:ServiceList"/>
</wsdl:message>
```

releaseService Messages

Request and response messages for the `releaseService` operation.

Syntax

```
<wsdl:message name="releaseServicesRequest">
   <wsdl:part name="services" type="tns1:ServiceList"/>
</wsdl:message>

<wsdl:message name="releaseServicesResponse">
   <wsdl:part name="releaseServicesResult" type="tns1:ServiceList"/>
</wsdl:message>
```

browseService Messages

Request and response messages for the `browseService` operation.

Syntax

```
<wsdl:message name="browseServiceRequest"></wsdl:message>

<wsdl:message name="browseServiceResponse">
   <wsdl:part name="browseServicesResult" type="tns1:ServiceList"/>
</wsdl:message>
```

getSessionTimeout Messages

Request and response messages for the `getSessionTimeout` operation.

Syntax

```
<wsdl:message name="getSessionTimeoutRequest"></wsdl:message>

<wsdl:message name="getSessionTimeoutResponse">
   <wsdl:part name="getSessionTimeoutResult" type="xsd:int"/>
</wsdl:message>
```

Operations

An operation is a combination of messages that, when successfully exchanged, perform a specific function. The Session Service includes six operations, each consisting of a simple pair of request and response messages.

login Operation

Authenticates the user for a session by verifying the user name and password, and provides the Session ID that is required for all other Session operations.

Syntax

```
<wsdl:operation name="login" parameterOrder="identity">
   <wsdl:input message="impl:loginRequest" name="loginRequest"/>
   <wsdl:output message="impl:loginResponse" name="loginResponse"/>
</wsdl:operation>
```

logout Operation

Terminates the user's session.

Syntax

```
<wsdl:operation name="logout" parameterOrder="sessionID">
   <wsdl:input message="impl:logoutRequest" name="logoutRequest"/>
   <wsdl:output message="impl:logoutResponse"
      name="logoutResponse"/>
</wsdl:operation>
```

getServices Operation

Checks out a license for the specified GIS SDK Service.

Syntax

```
<wsdl:operation name="getServices" parameterOrder="services">
   <wsdl:input message="impl:getServicesRequest"
      name="getServicesRequest"/>
   <wsdl:output message="impl:getServicesResponse"
      name="getServicesResponse"/>
</wsdl:operation>
```

releaseServices Operation

Releases a checked out license for the specified GIS SDK Service.

Syntax

```
<wsdl:operation name="releaseServices" parameterOrder="services">
   <wsdl:input message="impl:releaseServicesRequest"
      name="releaseServicesRequest"/>
   <wsdl:output message="impl:releaseServicesResponse"
      name="releaseServicesResponse"/>
</wsdl:operation>
```

browseServices Operation

Returns a list of the available licenses. You must have a service license checked out to use this operation.

Syntax

```
<wsdl:operation name="browseServices">
   <wsdl:input message="impl:browseServicesRequest"
      name="browseServicesRequest"/>
   <wsdl:output message="impl:browseServicesResponse"
      name="browseServicesResponse"/>
</wsdl:operation>
```

getSessionTimeout Operation

Returns the timeout value.

Syntax

```
<wsdl:operation name="getSessionTimeout">
   <wsdl:input message="impl:getSessionTimeoutRequest"
      name="getSessionTimeoutRequest"/>
   <wsdl:output message="impl:getSessionTimeoutResponse"
      name="getSessionTimeoutResponse"/>
</wsdl:operation>
```

Bindings

The name of the Session Service binding is
SessionServiceServiceSoapBinding. The binding style is RPC over HTTP.

Login Binding

SOAP binding for the `login` operation.

Syntax

```
<wsdl:operation name="login">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="loginRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="loginResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

Logout Binding

SOAP binding for the `logout` operation.

Syntax

```
<wsdl:operation name="logout">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="logoutRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="logoutResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

getServices Binding

SOAP binding for the `getServices` operation.

Syntax

```
<wsdl:operation name="getServices">
```

```
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="getServicesRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="getServicesResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

releaseServices Binding

SOAP binding for the `releaseServices` operation.

Syntax

```
<wsdl:operation name="releaseServices">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="releaseServicesRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="releaseServicesResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

browseServices Binding

SOAP binding for the `browseServices` operation.

Syntax

```
<wsdl:operation name="browseServices">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="browseServicesRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="browseServicesResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
```

```
      namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
      use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

getSessionTimeout Binding

SOAP binding for the `getSessionTimeout` operation.

Syntax

```
<wsdl:operation name="getSessionTimeout">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="getSessionTimeoutRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="getSessionTimeoutResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/sessionservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

Service

The Session Service service name is `SessionServiceService`.

Syntax

```
<wsdl:service name="SessionServiceService">
   <wsdl:port binding="impl:SessionServiceServiceSoapBinding"
      name="SessionServiceService">
   <wsdlsoap:address location="http://<GIS_HOST>:<GIS_PORT>/gis/services/
      SessionService"/>
   </wsdl:port>
</wsdl:service>
```

3
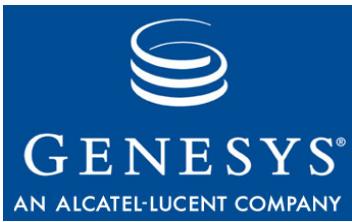
View Configuration Methods

You can use the methods documented in this chapter to access configuration data located in the Genesys Configuration Database.

This chapter assumes that you only have a Statistics SDK Service license, which means that your access is read-only. If you also have a Configuration SDK Service license, see the *Configuration SDK 7.6 Web Services API Reference* and *Configuration SDK 7.6 Developer's Guide* for complete instructions on using the Configuration SDK Service.

This chapter includes the following sections:

Overview

Genesys provides read-only access to configuration data for users with Statistics SDK Service licenses who do not also purchase Configuration SDK Service licenses. This chapter describes the methods available to retrieve configuration data.

The 7.*x* version of the Genesys Statistics SDK Service accesses configuration data in the same way as the Genesys Configuration SDK Service, by tunneling via GIS to Configuration Server's SOAP interface.

Note: The `RetrieveConfiguration` method that was used in versions 6.1 and 6.5.0 of the Genesys Statistics API is no longer available.

Available Methods

You can retrieve configuration information using these methods:

If you try to use other methods without having the Configuration SDK Service license, a `Service Not Allowed` error message is returned.

The Register Operation

This operation includes request and response messages for the `register` operation, which logs your client in to the Configuration Server:

- `CS.register`
- `CS.registerResponse`

You must use this operation as the first request from your application to the Configuration Server. The response includes a cookie, which HTTP uses to link the various messages into a single HTTP session.

Configuring Cookie Lifespan

The cookie lifespan determines the amount of time that your HTTP connection keeps information about a closed SOAP connection. You can configure the length of this setting during SOAP port configuration.

For details, see the section about configuring the Configuration Server SOAP port in your *Genesys Integration Server 7.6 Deployment Guide*.

Syntax

```
<gsd:register xmlns:gsd="http://genesyslab.com/cs/message/" SOAP-ENV:encodingStyle="">
   <userName SOAP-ENV:encodingStyle="">default</userName>
   <userPassword SOAP-ENV:encodingStyle="">password</userPassword>
   <applicationName SOAP-ENV:encodingStyle="">default</applicationName>
</gsd:register>

<gsd:registerResponse xmlns:gsd="http://genesyslab.com/cs/message/" SOAP-
   ENV:encodingStyle="">
  <result>OK</result>
</gsd:registerResponse>
```

The Get Operation

This operation includes the request and response messages for the `get` operation, which executes a query to retrieve configuration object data:

- `CS.get`
- `CS.getResponse`

Your request uses a Genesys-specific query language (based on a subset of XPath) to submit a query. For details about creating queries, see the *Configuration SDK 7.6 Web Services API Reference*'s "Creating Queries" chapter.

The response is in XML format with native DBID values and no link resolution. This means that the response returns precisely what was requested without the need for additional processing—as opposed to the results returned from a `getex` query. For more on `getex`, see "The Getex Operation" on page 35.

Note: The code snippet below uses sample values for the content of the query and the response. The content of your queries and responses depends on the information you are retrieving and the details of your environment.

Syntax

```
<gsd:get xmlns:gsd="http://genesyslab.com/cs/message/" SOAP-ENV:encodingStyle="">
   <query SOAP-ENV:encodingStyle="">CfgApplication[@name = 'default']</query>
</gsd:get>

<gsd:getResponse xmlns:gsd="http://genesyslab.com/cs/message/"
     SOAP-ENV:encodingStyle="">
   <result>
      <CfgData>
         <CfgApplication DBID='100' name='default' state='1'/>
      </CfgData>
   </result>
</gsd:getResponse>
```

The Getex Operation

This operation consists of request and response messages for the `getex` operation:
- `CS.getex`
- `CS.getexResponse`

This operation executes a query to retrieve a set of configuration object data in XML format with all link objects resolved. The `getex` operation differs from the `get` operation by returning values for a set of interconnected objects in a single step. For example, you can use the `getex` operation to retrieve data on an application and its backup server and its host in a single query.

For a comparison with the `get` operation, see "The Get Operation" on page 34. For details about creating valid queries, see the *Configuration SDK 7.6 Web Services API Reference*'s "Creating Queries" chapter.

Note: The code snippet below uses sample values for the content of the query and the response. The content of your queries and responses depends on the information you are retrieving and the details of your environment.

Syntax

```
<gsd:getex xmlns:gsd="http://genesyslab.com/cs/message/" SOAP-ENV:encodingStyle="">
   <query SOAP-ENV:encodingStyle="">CfgApplication[@DBID=110]</query>
   <isRecursive SOAP-ENV:encodingStyle="">true</isRecursive>
</gsd:getex>

<gsd:getexResponse xmlns:gsd="http://genesyslab.com/cs/message/"
     SOAP-ENV:encodingStyle="">
   <CfgData>
     <CfgCreate>
        <CfgApplication  id='CfgApp001'  name='TServer-001' switchDBID='CfgSwitch101'
           state='1' />
        <CfgSwitch  id='CfgSwitch101'  name='SwitchX'  ownerDBID='CfgProviderTenant1'
           state='1' />
     </CfgCreate>
     <CfgReference>
        <CfgProviderTenantRef id='CfgProviderTenant1' name='Environment'/>
     </CfgReference>
   </CfgData>
</gsd:getexResponse>
```

The GetVersion Operation

This operation consists of request and response messages for the `getVersion` operation:

- `CS.getVersion`
- `CS.getVersionResponse`

This operation retrieves the version of Configuration Server to which your application is connected.

Syntax

```
<gsd:getVersion xmlns:gsd="http://genesyslab.com/cs/message/"
   SOAP-ENV:encodingStyle="">
</gsd:getVersion>

<gsd:getVersionResponse xmlns:gsd="http://genesyslab.com/cs/message/"
     SOAP-ENV:encodingStyle="">
  <serverVersion>7.0.000.01</server Version>
</gsd:getVersionResponse>
```

4

Statistics SDK Service Namespaces

This chapter lists the standard and Genesys-specific namespaces you need to create statistics-gathering messages.

This chapter includes the following section:

Namespaces

The Statistics SDK Service uses the namespaces described in Table 2.

Table 2: Statistics WSDL Namespaces

Namespace	Description
`targetNamespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"`	Genesys-specific namespace
`xmlns:impl="http://www.genesyslab.com/services/statservice/wsdl/2002/03"`	Genesys-specific namespace
`xmlns:intf="http://www.genesyslab.com/services/statservice/wsdl/2002/03"`	Genesys-specific namespace
`xmlns:tns1="http://www.genesyslab.com/services/statservice/types/2005/10"`	Namespace for the Genesys type definitions
`xmlns:tns2="http://www.genesyslab.com/services/statservice/types/2002/03"`	Namespace for the Genesys type definitions

Table 2: Statistics WSDL Namespaces (Continued)

Namespace	Description
`xmlns:xsd="http://www.w3.org/2001/XMLSchema"`	Namespace for the XML Schema definitions
`xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"`	Namespace for the SOAP definitions
`xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"`	Genesys-specific namespace
`xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"`	Namespace for the WSDL definitions
`xmlns:apachesoap="http://xml.apache.org/xml-soap"`	Genesys-specific namespace

![GENESYS AN ALCATEL-LUCENT COMPANY]

Chapter

5

Statistics SDK Service Types

Types (or data types) define the structure of information that is contained in a message. This chapter describes the two types: simple and complex.

This chapter includes the following sections:

- Simple Types Summary, page 39
- Complex Types Summary, page 40
- Simple Types in Detail, page 43
- Complex Types in Detail, page 53

Simple Types Summary

Simple types consist of simple data types including integers, strings, and enumeration values, as explained in Table 3.

Table 3: Simple Types

Simple Type	Description
"eventValueTypeType" on page 44	The possible statistic value types: `float`, `Long`, or `unknownValue`.
"notificationMode" on page 44	Specifies whether GIS should use the Polling or Blocked mode to update statistics. **Note:** Unsolicited Notification mode uses an alternate method. See "unsolicitedNotification" on page 70 for details.
"objectType" on page 46	A statistical object type, such as `Agent`, `RoutePoint`, or `GroupPlaces`.

Table 3: Simple Types (Continued)

Simple Type	Description
"scheduleMode" on page 48	The type of event that prompts a statistic update, such as `ChangesBased` or `ResetBased`.
"statisticalProfileType" on page 50	A statistical profile type such as `timeProfile` or `filterProfile`.
"statisticStateType" on page 51	The status for a specified object, such an agent or a DN.
"timeIntervalType" on page 52	A time interval type such as `SlidingWindow` or `GrowingWindow`.

Complex Types Summary

Complex types consist of sequences or combinations of simple types and Genesys type definitions, as described in Table 4.

Table 4: Complex Types

Complex Type	Description	Element Name	Element Type
"actions" on page 53	A list of actions for a statistical type (stat type).	action	xsd:string
"ArrayOf_subscription" on page 55	An array of elements of a specified type.	subscription	tns1:statistic Subscription[]
"eventValue" on page 57	A statistical value.	date	xsd:long
		intervalLength	xsd:long
		fValue	xsd:float
		lValue	xsd:long
		sValue	xsd:string
		stateValue	Not currently used—the value is always `null`.
"eventValues" on page 58	An array of event values.	eventValues	tns1:eventValue[]

Table 4: Complex Types (Continued)

Complex Type	Description	Element Name	Element Type
"metric" on page 59	A statistical metric.	typeName	xsd:string
		statisticType	tns1:statisticType
		interval	tns1:timeInterval
		filterName	xsd:string
		timeRangeName	xsd:string
		timeRange	tns1:timeRangeType
"notification" on page 60	Notification modes.	mode	tns1:notificationMode
		timeout	xsd:int
"objectIdType" on page 61	A list of object types.	id	xsd:string
		tenantName	xsd:string
"objectTypes" on page 61	An array of objectType values.	objectType	tns1:objectType[]
"parameter" on page 61	A key-value pair.	key	xsd:string
		value	xsd:string
"parameters" on page 62	A list of parameters.	parameter	tns1:parameter[]
"profileInfo" on page 62	A statistical profile.	profileType	tns1:statisticalProfileType
		statisticInfos	tns1:statisticInfos
		timeProfiles	tns1:timeProfiles
		filters	tns1:parameters
		timeRanges	tns1:parameters
"retrieveStatisticalProfileResponse" on page 62	A response to a request to retrieve a statistical profile.	statisticalProfileInfo	tns1:profileInfo
"retrieveStatisticResponse" on page 63	A response to a request to retrieve a statistic.	statisticValue	tns1:statisticValue

Table 4: Complex Types (Continued)

Complex Type	Description	Element Name	Element Type
"retrieveSubscribedStatistics Response" on page 63	A response to the client's request to subscribe to statistics.	statisticValues	tns1:statisticValues
"schedule" on page 63	A notification schedule.	notification Mode	tns1:scheduleMode
		timeout	xsd:int
		insensitivity	xsd:int
"statistic" on page 64	A statistic.	statisticId	xsd:string
		objectId	tns1:objectIdType
		metric	tns1:metric
		schedule	tns1:schedule
"statisticInfos" on page 65	An array of statisticTypeInfoType values.	statisticTypeInfo Type	tns1:statisticTypeInfo Type[]
statisticStateData	Not currently used.		
statisticState	Not currently used.		
statisticStates	Not currently used.		
"statisticSubscription" on page 66	A statistic subscription.	scope	xsd:string
		statisticId	xsd:string
"statisticSubscriptions" on page 66	A list of statistic subscriptions.	statistic Subscription	tns1:ArrayOf_subscript ion
"statisticType" on page 67	The definition of a statistical type.	objectType	tns1:objectTypes
		category	xsd:string
		subject	xsd:string
		mainActions	tns1:actions
		relativeActions	tns1:actions

Table 4: Complex Types (Continued)

Complex Type	Description	Element Name	Element Type
"statisticTypeInfoType" on page 68	The type of statistical information in the message.	typeName	xsd:string
		type	tns1:eventValueType Type
		objectTypes	tns1:objectTypes
"statisticValue" on page 68	A statistical value.	statisticId	xsd:string
		eventValues	tns1:eventValues
"statisticValues" on page 68	An array of statistic values.	statisticValues	tns1:statisticValue[]
"timeInterval" on page 69	A statistical time interval.	intervalType	tns1:timeIntervalType
		length	xsd:int
		slideLength	xsd:int
		timeProfile Name	xsd:string
"timeProfile" on page 69	A statistical time profile.	key	xsd:string
		intervalType	tns1:timeIntervalType
		value	xsd:string
"timeProfiles" on page 69	An array of statistical time profiles.	timeProfile	tns1:timeProfile[]
"timeRangeType" on page 70	A statistical time range.	leftTime	xsd:int
		rightTime	xsd:int
"unsolicitedNotification" on page 70	Specifies the URL to which GIS sends automatic notification of configuration updates.	url	xsd:string

Simple Types in Detail

This section explains how to use each simple type, the enumeration values that are available for each simple type, and what each enumeration value signifies.

eventValueTypeType

Use this type to specify the value type for each statistic. Figure 9 on page 81 shows an example of a message that includes the `eventValueTypeType` type.

Syntax

```
<simpleType name="eventValueTypeType">
   <restriction base="xsd:string">
      <enumeration value="fValue"/>
      <enumeration value="lValue"/>
      <enumeration value="unknownValue"/>
   </restriction>
</simpleType>
```

Table 5 lists the enumeration values for `eventValueTypeType` and their definitions.

Table 5: eventValueTypeType Enumeration Values

Enumeration Value	Description
fValue	The statistical value type is float.
lValue	The statistical value type is long.
unknownValue	The statistical value type cannot be determined.

notificationMode

Use this type to indicate which solicited notification mode to use. Figure 6 on page 78 shows an example of a message that includes the `notificationMode` type.

Syntax

```
<simpleType name="notificationMode">
   <restriction base="xsd:string">
      <enumeration value="Polling"/>
      <enumeration value="Blocked"/>
   </restriction>
</simpleType>.
```

Table 6 lists the enumeration values for `notificationMode` and their definitions.

Table 6: notificationMode Enumeration Values

| Enumeration Value	Description
Polling	GIS returns an immediate response with the current statistical value. Polling notification mode uses the following steps: 1. HTTP connection opened. 2. Client sends request. 3. Server sends response. 4. HTTP connection closed. The timeout is the time interval after which the GIS sends a fault response if no events are sent by the server.

Blocked	Responds when the next updated statistical value is available. Blocking notification mode uses the following steps, which are also displayed in Figure 1 on page 46:
	1. HTTP connection opened.
	2. Client sends request.
	3. Client is blocked (HTTP connection remains open).
	4. Client is unblocked (HTTP connection is closed) when:
	    a. The Server sends a response (an event). In this scenario, the event is raised before the end of the timeout value.
	    b. The timeout period expires before a response (an event) arrives.
	If you set the timeout value to -1, the client waits indefinitely for a server response.
	Due to the HTTP nature of the request, a timeout can expire on client side. The next request to retrieve the same statistic will throw a timeout exception until a new statistic event arrives from Stat Server.
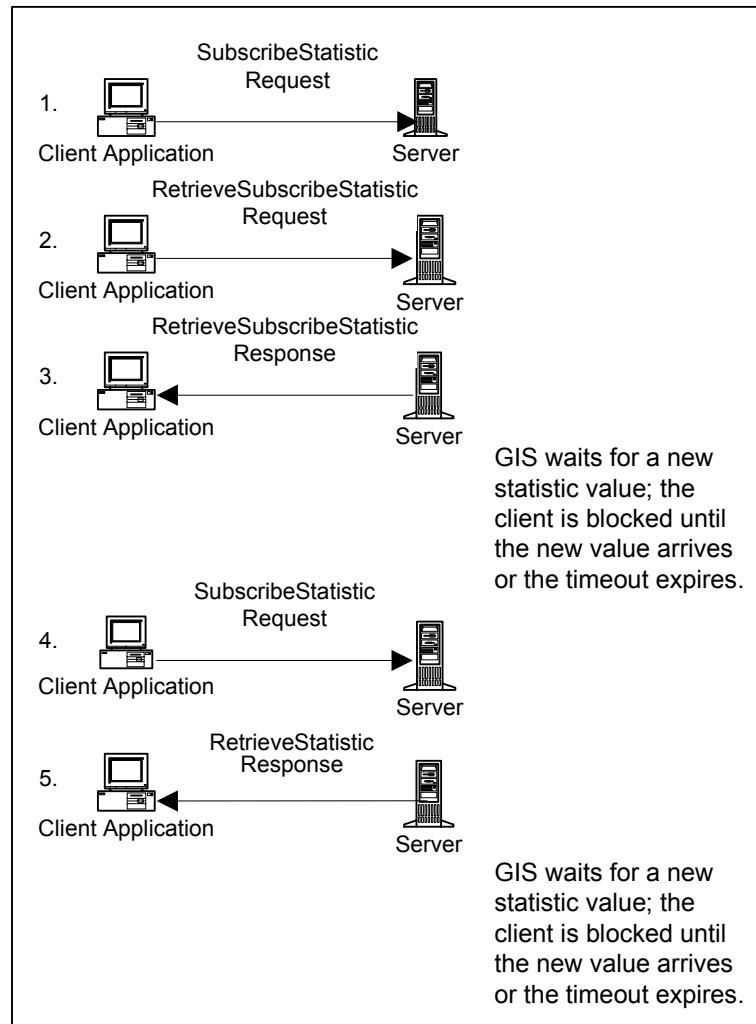
Figure 1: Blocked Notification Flow

objectType

Use this type to specify the object types the statistic monitors.

Figure 3 on page 75, Figure 9 on page 81, and Figure 5 on page 77 show examples of a message that includes the objectType type.

Syntax

```
<simpleType name="objectType">
   <restriction base="xsd:string">
      <enumeration value="Agent"/>
      <enumeration value="CallingList"/>
      <enumeration value="Campaign"/>
      <enumeration value="CampaignCallingList"/>
      <enumeration value="CampaignGroup"/>
```

```
        <enumeration value="GroupAgents"/>
        <enumeration value="GroupPlaces"/>
        <enumeration value="GroupQueues"/>
        <enumeration value="Place"/>
        <enumeration value="Queue"/>
        <enumeration value="RegDN"/>
        <enumeration value="RoutePoint"/>
        <enumeration value="StagingArea"/>
        <enumeration value="Strategy"/>
        <enumeration value="Tenant"/>
    </restriction>
</simpleType>
```

Table 7 lists the enumeration values for `objectType` and their definitions.

Table 7: objectType Enumeration Values

Enumeration Value	Description
Agent	An individual identified by an Agent ID and defined as an Employee ID in the Configuration Manager. At any given time, each agent can be at only one place and each place can be occupied by only one agent.
Campaign	Campaign statistics are calculated exclusively for Outbound Contact Solution to reflect campaign performance.
CallingList	
CampaignGroup	A specific `CampaignGroup` object is based on an `Agent Group` object assigned to a specific campaign, and a specific `Campaign CallingList` object is based on a `CallingList` object assigned to a specific campaign.
CampaignCallingList	
	`CampaignGroup` and `CampaignCallingList` objects must be named `campaign@groupname` and `campaign@callinglist`, respectively, where `groupname (callinglist)` is the name of a specific `AgentGroup (CallingList)` assigned to the `campaign` and can be viewed in CCPulse+ under these names.
GroupAgents	A group of agents identified by a Group ID. An agent can be a member of more than one group.
GroupPlaces	A group of places. Each place that is part of the group has a unique Place ID that is associated with the Group ID.
GroupQueues	A group that includes queues, route points, virtual queues, and virtual route points.

Table 7: objectType Enumeration Values (Continued)

Enumeration Value	Description
Place	A location identified by a Place ID. Even if various agents move in and out of a place, the activities that occur at the place can be tracked by the Place ID.
Queue	ACD-associated points at which calls wait for the next available agent.
RegDN	RegDN applies to the following DN types: data, music, mixed, extension, ACD position, Voice Treatment port, voice mail, cellular, CP (call-processing equipment), and fax. Except for extensions and Voice Treatment ports, all these DN types require login events.
RoutePoint	A point on a switching platform where calls wait until they are routed. Route points have different names on different switching platforms (for example, CDN or VDN).
StagingArea	Analogous to the concept of queues for the Multi-Channel Routing (MCR) solution, where customer interactions may reside while being processed by MCR. This object type corresponds to the Script Configuration Server type, StagingArea subtype.
Strategy	A routing strategy deployed by the Genesys Interaction Routing Designer tool. The strategy is manifested in Configuration Server as a Script object of type Simple Strategy or Strategy.
Tenant	A Tenant represents a business entity within the Configuration Server.

scheduleMode

Use this type to specify the kind of event that GIS should consider an update.

Figure 3 on and Figure 5 on show examples of a message that includes the scheduleMode type.

Syntax

```
<simpleType name="scheduleMode">
  <restriction base="xsd:string">
    <enumeration value="ChangesBased"/>
    <enumeration value="TimeBased"/>
```

```
          <enumeration value="ResetBased"/>
      </restriction>
</simpleType>
```

Table 8 lists the enumeration values for `scheduleMode` and their definitions.

Table 8: scheduleMode Enumeration Values

Enumeration Value	Description

ChangesBased	Notification received when a statistical value changes. Used with the `insensitivity` parameter.
	• To be notified of all changes, set `insensitivity` equal to zero (= `0`).
	• To be notified of more significant changes, set `insensitivity` using the following formula:
	`abs(NewValue - LastSentValue) * 100 / LastSentValue >`
	`Insensitivity Value`
	• Can be used with all time interval types.

ResetBased	Notification received according to the time points specified with `timeProfile`. The statistic is reset to zero (if not current). • Can be used with the `GrowingWindow` interval type.
TimeBased	Notification received after *N* seconds, regardless of changes. • Can be used with all time interval types.

Figure 2 shows how `scheduleMode` enumeration values interact with `timeIntervalType` values. To ensure a valid configuration, select a combination marked in Figure 2 with a black dot.

For details about the `timeIntervalType` type, see "timeIntervalType" on
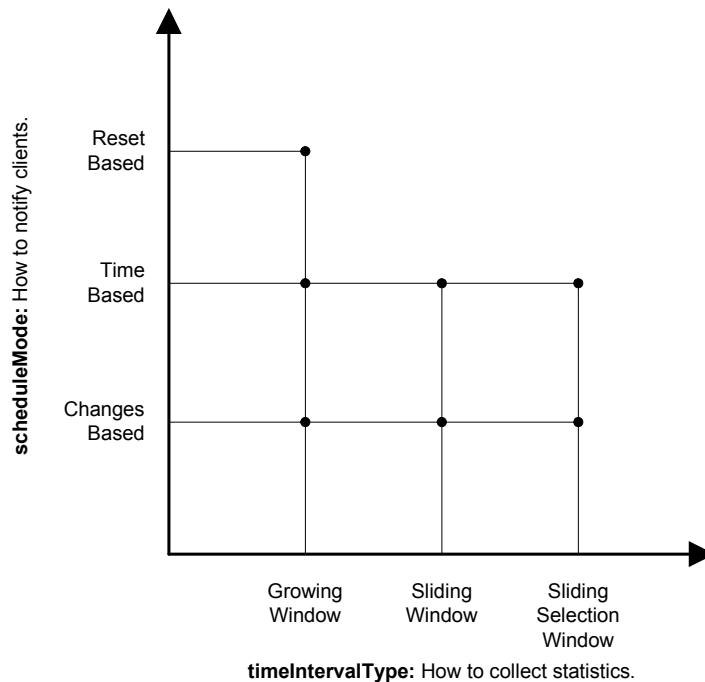
Figure 2: Collecting Historical Statistics

statisticalProfileType

Use this type to specify the statistical profile that is used.

Figure 8 on page 80 and Figure 9 on page 81 show examples of a message that includes the statisticalProfileType type.

Syntax

```
<simpleType name="statisticalProfileType">
   <restriction base="xsd:string">
      <enumeration value="statisticalProfile"/>
      <enumeration value="timeProfile"/>
      <enumeration value="filterProfile"/>
      <enumeration value="timeRangeProfile"/>
   </restriction>
</simpleType>
```

Table 9 lists the enumeration values for statisticalProfileType and their definitions.

Table 9: statisticalProfileType Enumeration Values

Enumeration Value	Description
filterProfile	Excludes calls based on certain criteria specified in a logical condition. Filters restrict the Stat Server actions taken into account in the computation of aggregate values.
statisticalProfile	Statistical types (stat types). Defines statistical retrievals based on a set of configuration options: objects, category, mainmask, relmask, subject, and formula.
timeProfile	Defines time intervals used to calculate historical aggregate values for statistics. See "timeIntervalType" on page 52.
timeRangeProfile	Defines a time range for collecting data. The time-range name is any character string that represents the time range. The time range value is composed of two digits, the start and end points, separated by a hyphen, for example, 0-20 seconds.

statisticStateType

Use this type to specify the status of an object such as an agent or DN.

Figure 4 on page 76 shows an example of a message that includes the statisticalProfileType type.

Syntax

```
<simpleType name="statisticStateType">
   <restriction base="xsd:string">
      <enumeration value="DNAction" />
      <enumeration value="DNStatus" />
      <enumeration value="AgentStatus" />
      <enumeration value="GroupStatus" />
   </restriction>
</simpleType>
```

timeIntervalType

Use this type to specify the time interval type for collecting statistical values. Figure 2 on page 50 presents the interaction between `scheduleMode` and `timeIntervalType`.

Figure 3 on page 75, Figure 9 on page 81, and Figure 5 on page 77 show example of a message that includes the `statisticalProfileType` type.

Syntax

```
<simpleType name="timeIntervalType">
   <restriction base="xsd:string">
      <enumeration value="GrowingWindow"/>
      <enumeration value="SlidingWindow"/>
      <enumeration value="SlidingSelectionWindow"/>
   </restriction>
</simpleType>
```

Table 10 lists the enumeration values for `timeIntervalType` and their definitions.

Table 10: timeIntervalType Enumeration Values

Enumeration Value	Description
GrowingWindow	Fixed start time, sliding end time ("until now"). For example, the total number of calls from the beginning of a shift (8:00 AM) until now.
SlidingSelectionWindow	Event-specific window. Sliding start time, sliding end time based on the occurrence of specific actions (listed in the Relative Mask or End Mask). For example, the total call length for the last seven calls.
SlidingWindow	Time-specific window. Sliding start time, sliding end time, fixed length. For example, the average call length for the last 10 minutes.

Complex Types in Detail

Some of the type definitions in this section include a status table. The meaning of the status designations are:

- **R—Required:** the value of the element must be present.
- **E—Empty:** The value of the element can be the empty string.
- **O—Omitted:** The value of the element can be omitted.

actions

A list of actions for a statistical type.

Syntax

```
<complexType name="actions">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns1:stateDNAction[]" />
      </restriction>
   </complexContent>
</complexType>
```

actions

A list of actions for a statistical type.

- **Element**—action
- **Type**—string
- **Status**—R

To designate an action, use the action name. To prevent an action from being used, use the tilde symbol (~) with the action name. For example:

- CallInterval—Use the CallInterval action.
- ~CallInterval—Do not use the CallInterval action.

For example, the statistic `totalLoginTime` is defined below. The MainMask, which corresponds to the list of actions, states to use all actions except `LoggedOut` and `NotMonitored`.

```
Category=TotalTime
MainMask=*, ~LoggedOut, ~NotMonitored
Objects=Agent, GroupAgents
RelMask=LoggedOut
Subject=AgentStatus
```

Examples of this complex type being used in a request are given in Figure 3 on page 75 and Figure 5 on page 77.

Syntax

```
<complexType name="actions">
   <sequence>
      <element name="action" nillable="true" type="impl:ArrayOf_xsd_string"/>
   </sequence>
</complexType>
```

agents

A array of elements of the type "agentStatus".

Syntax

```
<complexType name="agents">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns1:agentStatus[]" />
      </restriction>
   </complexContent>
</complexType>
```

agentReasons

A listing of agent reasons.

Syntax

```
<complexType name="agentReasons">
   <sequence>
      <element name="agentId" nillable="true" type="xsd:string" />
      <element name="loginId" nillable="true" type="xsd:string" />
      <element name="placeId" nillable="true" type="xsd:string" />
      <element name="reasons" nillable="true" type="tns1:kvListValue" />
      <element name="status" nillable="true" type="xsd:string" />
      <element name="tmStart" type="xsd:long" />
   </sequence>
</complexType>
```

agentStatus

The status of a given agent.

Syntax

```
<complexType name="agentStatus">
   <sequence>
      <element name="agentId" nillable="true" type="xsd:string" />
      <element name="dns" nillable="true" type="tns1:dns" />
      <element name="loginId" nillable="true" type="xsd:string" />
      <element name="placeId" nillable="true" type="xsd:string" />
      <element name="status" nillable="true" type="xsd:string" />
      <element name="tmStart" type="xsd:long" />
   </sequence>
</complexType>
```

ArrayOf_subscription

An array of elements of the type "timeProfile".

An example of this complex type being used in a request is given in Figure 6 on .

Syntax

```
<complexType name="ArrayOf_subscription">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType"
            wsdl:arrayType="tns2:statisticSubscription[]"/>
      </restriction>
   </complexContent>
</complexType>
```

ArrayOf_xsd_string

An array of elements of the type "xsd:string".

Syntax

```
<complexType name="ArrayOf_xsd_string">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]" />
      </restriction>
   </complexContent>
</complexType>
```

campaignStatus

The status of a given campaign.

Syntax

```
<complexType name="campaignStatus">
   <sequence>
      <element name="campaignDBID" nillable="true" type="xsd:string" />
      <element name="campaignID" nillable="true" type="xsd:string" />
      <element name="dnAction" nillable="true" type="tns1:actions" />
      <element name="status" nillable="true" type="xsd:string" />
      <element name="tmStart" type="xsd:long" />
   </sequence>
</complexType>
```

connID

A connection ID.

Syntax

```
<complexType name="connID">
   <sequence>
      <element name="conn_id" nillable="true" type="xsd:base64Binary" />
   </sequence>
</complexType>
```

dataCall

A listing of call data.

Syntax

```
<complexType name="dataCall">
   <sequence>
      <element name="connId" nillable="true" type="tns1:connID" />
      <element name="sani" nillable="true" type="xsd:string" />
      <element name="sdnis" nillable="true" type="xsd:string" />
      <element name="userData" nillable="true" type="tns1:kvListValue" />
   </sequence>
</complexType>
```

dns

An array of type "dnStatus".

Syntax

```
<complexType name="dns">
   <complexContent>
```

```
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns1:dnStatus[]" />
      </restriction>
   </complexContent>
</complexType>
```

dnStatus

The status of a given DN.

Syntax

```
<complexType name="dnStatus">
   <sequence>
      <element name="dnAction" nillable="true" type="tns1:actions" />
      <element name="dnId" nillable="true" type="xsd:string" />
      <element name="status" nillable="true" type="xsd:string" />
      <element name="switchId" nillable="true" type="xsd:string" />
      <element name="tmStart" type="xsd:long" />
      <element name="type" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

eventValue

A statistical value.

Examples of this complex type being used in a request are given in Figure 4 on page 76 and Figure 7 on page 79.

Syntax

```
<complexType name="eventValue">
   <sequence>
      <element name="FValue" type="xsd:float" />
      <element name="LValue" type="xsd:long" />
      <element name="SValue" nillable="true" type="xsd:string" />
      <element name="agentReasons" nillable="true" type="tns1:agentReasons" />
      <element name="agentStatus" nillable="true" type="tns1:agentStatus" />
      <element name="campaignStatus" nillable="true" type="tns1:campaignStatus" />
      <element name="date" type="xsd:long" />
      <element name="dnStatus" nillable="true" type="tns1:dnStatus" />
      <element name="groupStatus" nillable="true" type="tns1:groupStatus" />
      <element name="intervalLength" type="xsd:long" />
      <element name="stateValue" nillable="true" type="tns2:statisticState" />
   </sequence>
</complexType>
```

eventValues

An array of `eventValue` items.

Examples of this complex type being used in a request are given in Figure 4 on and Figure 7 on .

Syntax

```
<complexType name="eventValues">
   <complexContent>
     <restriction base="soapenc:Array">
        <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:eventValue[]" />
     </restriction>
   </complexContent>
</complexType>
```

groupStatus

The status of a given group.

Syntax

```
<complexType name="groupStatus">
   <sequence>
     <element name="agents" nillable="true" type="tns1:agents" />
     <element name="groupId" nillable="true" type="xsd:string" />
     <element name="status" nillable="true" type="xsd:string" />
     <element name="tmStart" type="xsd:long" />
   </sequence>
</complexType>
```

keyValue

A keyValue.

Syntax

```
<complexType name="keyValue">
   <sequence>
     <element name="binaryValue" nillable="true" type="xsd:base64Binary" />
     <element name="intValue" type="xsd:int" />
     <element name="key" nillable="true" type="xsd:string" />
     <element name="kvListValue" nillable="true" type="tns1:kvListValue" />
     <element name="stringValue" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

kvListValue

A listing of "keyValue".

Syntax

```
<complexType name="kvListValue">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns1:keyValue[]" />
      </restriction>
   </complexContent>
</complexType>
```

metric

A statistical metric as shown in Table 11. There are two ways to subscribe to statistics:

1. Use `typeName` to specify predefined stat types. The Stat Server will retrieve all the parameters automatically. For example, request `typeName = "CurrentLogoutNumber"`, `CurrentLogoutNumber` being a statistic type defined in the `StatTypes.cfg` file.

2. Use `statisticType` to manually specify all the parameters required by the Stat Server. For example, configure you client application to use `statisticType` to define the `objectType`, `category`, `subject`, and `mainAction` for a statistic.

Examples of this complex type being used in a request are given in Figure 3 on page 75 and Figure 5 on page 77.

Table 11: Metric Type

Element	Type	Status
typeName	xsd:string	O
statisticType	tns1:statisticType	O
interval	tns1:timeInterval	R
filterName	xsd:string	E, O
timeRangeName	xsd:string	E, O
timeRange	tns1:timeRangeType	O

•   Use either `typeName` or `statisticType` but not both.

•   Use either `timeRangeName` or `timeRange` but not both.

- If you use `typeName`, use `timeRangeName`. If you use `statisticType`, use `timeRange`.

Syntax

```
<complexType name="metric">
   <sequence>
      <element name="filterName" nillable="true" type="xsd:string" />
      <element name="interval" nillable="true" type="tns2:timeInterval" />
      <element name="statisticType" nillable="true" type="tns2:statisticType" />
      <element name="timeRange" nillable="true" type="tns2:timeRangeType" />
      <element name="timeRangeName" nillable="true" type="xsd:string" />
      <element name="typeName" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

notification

The notification mode selected by the user.

> **Note:** Notification enables you to specify Polling or Blocked mode. To use unsolicited notification, you must enter your client application web server's URL in the `unsolicitedNotification` element. For details on unsolicited notification, see "unsolicitedNotification" on page 70.

In Polling mode, timeout is the interval after which the GIS sends a fault response to the client if no events are returned from the Stat Server.

If Blocked mode is selected, there are two cases:

- If timeout is not equal to -1, then the HTTP connection closes when an event is returned or the timeout expires.
- If timeout is equal to -1, then the HTTP connection remains open until a new event arrives or the client closes the connection.

An example of this complex type being used in a request is given in Figure 6 on page 78.

Syntax

```
<complexType name="notification">
   <sequence>
      <element name="mode" nillable="true" type="tns2:notificationMode" />
      <element name="timeout" nillable="true" type="xsd:integer" />
   </sequence>
</complexType>
```

objectIdType

A list of statistical object types.

Examples of this complex type being used in a request are given in Figure 3 on page 75 and Figure 5 on page 77.

Syntax

```
<complexType name="objectIdType">
   <sequence>
      <element name="id" nillable="true" type="xsd:string" />
      <element name="tenantName" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

objectTypes

An array of `objectType` items.

Examples of this complex type being used in a request are given in Figure 3 on page 75 and Figure 5 on page 77.

Syntax

```
<complexType name="objectTypes">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:objectType[]" />
      </restriction>
   </complexContent>
</complexType>
```

parameter

A key-value pair.

An example of this complex type being used in a request is given in Figure 9 on page 81.

Syntax

```
<complexType name="parameter">
   <sequence>
      <element name="key" nillable="true" type="xsd:string" />
      <element name="value" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

parameters

An array of `parameter` items.

An example of this complex type being used in a request is given in Figure 9 on page 81.

Syntax

```
<complexType name="parameters">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:parameter[]" />
      </restriction>
   </complexContent>
</complexType>
```

profileInfo

A statistical profile.

An example of this complex type being used in a request is given in Figure 9 on page 81.

Syntax

```
<complexType name="profileInfo">
   <sequence>
      <element name="profileType" nillable="true" type="tns2:statisticalProfileType" />
      <element name="statisticInfos" nillable="true" type="tns2:statisticInfos" />
      <element name="timeProfiles" nillable="true" type="tns2:timeProfiles" />
      <element name="filters" nillable="true" type="tns2:parameters" />
      <element name="timeRanges" nillable="true" type="tns2:parameters" />
   </sequence>
</complexType>
```

retrieveStatisticalProfileResponse

A response to a request to retrieve a statistical profile. The response returns a statistical profile.

An example of this complex type being used in a request is given in Figure 9 on page 81.

Syntax

```
<complexType name="retrieveStatisticalProfileResponse">
   <sequence>
      <element name="statisticalProfileInfo" nillable="true" type="tns2:profileInfo" />
```

```
   </sequence>
</complexType>
```

retrieveStatisticResponse

A response to a request to retrieve a statistic. The response returns a statistic.

An example of this complex type being used in a request is given in Figure 4 on .

Syntax

```
<complexType name="retrieveStatisticResponse">
   <sequence>
      <element name="statisticValue" nillable="true" type="tns2:statisticValue" />
   </sequence>
</complexType>
```

retrieveSubscribedStatisticsResponse

A response to a request for subscribed statistics. The response returns one or more statistics.

An example of this complex type being used in a request is given in Figure 7 on .

Syntax

```
<complexType name="retrieveSubscribedStatisticsResponse">
   <sequence>
      <element name="statisticValues" nillable="true" type="tns2:statisticValues" />
   </sequence>
</complexType>
```

schedule

A notification schedule, as shown in Table 12. There are three notification modes. The insensitivity element is used with the `ChangesBased` reset mode to determine how much a statistical value can change before notification is sent (that the value has changed). The timeout element refers to the time interval.

Chapter 5: Statistics SDK Service Types   Complex Types in Detail

Examples of this complex type being used in a request are given in Figure 3 on page 75 and Figure 5 on page 77.

Table 12: Schedule Type

Element	Type	Status
notificationMode	tns1:scheduleMode	R
timeout	xsd:int	O
insensitivity	xsd:int	O

Syntax

```
<complexType name="schedule">
   <sequence>
      <element name="insensitivity" nillable="true" type="xsd:int" />
      <element name="notificationMode" nillable="true" type="tns2:scheduleMode" />
      <element name="timeout" nillable="true" type="xsd:int" />
   </sequence>
</complexType>
```

stateDNAction

A stateDNAction.

Syntax

```
<complexType name="stateDNAction">
   <sequence>
      <element name="action" nillable="true" type="xsd:string" />
      <element name="data" nillable="true" type="tns1:dataCall" />
      <element name="tmStart" type="xsd:long" />
   </sequence>
</complexType>
```

statistic

A statistic, as detailed in Table 13.

Table 13: Statistic Type

Element	Type	Status
statisticId	xsd:string	R
objectId	tns1:objectIdType	R

64   Statistics SDK 7.6 Web Services

Table 13: Statistic Type

Element	Type	Status
metric	tns1:metric	R
schedule	tns1:schedule	O

See for Figure 3 on and Figure 5 on examples of this complex type being used in a request.

Syntax

```
<complexType name="statistic">
   <sequence>
      <element name="statisticId" nillable="true" type="xsd:string" />
      <element name="objectId" nillable="true" type="tns2:objectIdType" />
      <element name="metric" nillable="true" type="tns2:metric" />
      <element name="schedule" nillable="true" type="tns2:schedule" />
   </sequence>
</complexType>
```

statisticInfos

An array of `statisticTypeInfoType` items.

An example of this complex type being used in a request is given in Figure 9 on .

Syntax

```
<complexType name="statisticInfos">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType"
            wsdl:arrayType="tns2:statisticTypeInfoType[]" />
      </restriction>
   </complexContent>
</complexType>
```

statisticStateData

```
<complexType name="statisticStateData">
   <sequence>
      <element name="ANI" nillable="true" type="xsd:string" />
      <element name="DNIS" nillable="true" type="xsd:string" />
      <element name="agent" nillable="true" type="xsd:string" />
      <element name="dnType" nillable="true" type="xsd:string" />
      <element name="list" nillable="true" type="tns2:parameters" />
      <element name="loginID" nillable="true" type="xsd:string" />
```

```
      <element name="place" nillable="true" type="xsd:string" />
      <element name="switchID" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

statisticState

```
<complexType name="statisticState">
   <sequence>
      <element name="date" type="xsd:long" />
      <element name="extendedData" nillable="true" type="tns2:statisticStateData" />
      <element name="id" nillable="true" type="xsd:string" />
      <element name="items" nillable="true" type="tns2:statisticStates" />
      <element name="state" nillable="true" type="xsd:string" />
      <element name="type" nillable="true" type="tns2:statisticStateType" />
   </sequence>
</complexType>
```

statisticStates

```
<complexType name="statisticStates">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:statisticState[]" />
      </restriction>
   </complexContent>
</complexType>
```

statisticSubscription

A subscription to a statistic.

An example of this complex type being used in a request is given in Figure 6 on .

Syntax

```
<complexType name="statisticSubscription">
   <sequence>
      <element name="scope" nillable="true" type="xsd:string" />
      <element name="statisticId" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

statisticSubscriptions

A list of statistic that you have created subscriptions for.

An example of this complex type being used in a request is given in Figure 6 on .

Syntax

```
<complexType name="statisticSubscriptions">
   <sequence>
      <element name="statisticSubscription" nillable="true"
         type="tns2:ArrayOf_subscription"/>
   </sequence>
</complexType>
```

statisticType

A statistical type or stat type, as described in Table 14. For example, statistical type TotalLoginTime is defined as follows:

```
[TotalLoginTime]
Category=TotalTime
MainMask=*,~LoggedOut,~NonMonitored
Objects=Agent, GroupAgents
Subject=AgentStatus
```

Examples of this complex type being used in a request are given in Figure 3 on and Figure 5 on .

Table 14: StatisticType

Element	Type	Status
objectType	tns1:objectTypes	R
category	xsd:string	R
subject	xsd:string	R
mainActions	tns1:actions	R
relativeActions	tns1:actions	E, O

Note: For compatibility with the Stat Server statistical types, there can be more than one objectType within a sequence. However, the GIS only uses the first objectType.

Syntax

```
<complexType name="statisticType">
   <sequence>
```

```
      <element name="objectType" nillable="true" type="tns2:objectTypes" />
      <element name="category" nillable="true" type="xsd:string" />
      <element name="subject" nillable="true" type="xsd:string" />
      <element name="mainActions" nillable="true"
         type="tns2:actions" />
      <element name="relativeActions" nillable="true"
         type="tns2:actions" />
   </sequence>
</complexType>
```

statisticTypeInfoType

Indicates the nature of the statistical information included in a message.

An example of this complex type being used in a request is given in Figure 9 on .

Syntax

```
<complexType name="statisticTypeInfoType">
   <sequence>
      <element name="typeName" nillable="true" type="xsd:string" />
      <element name="type" nillable="true" type="tns2:eventValueTypeType" />
      <element name="objectTypes" nillable="true" type="tns2:objectTypes" />
   </sequence>
</complexType>
```

statisticValue

A statistical value based on the eventValues type.

Examples of this complex type being used in a request are given in Figure 4 on and Figure 7 on .

Syntax

```
<complexType name="statisticValue">
   <sequence>
      <element name="statisticId" nillable="true" type="xsd:string" />
      <element name="eventValues" nillable="true" type="tns2:eventValues" />
   </sequence>
</complexType>
```

statisticValues

An array of statisticValue items.

Examples of this complex type being used in a request are given in Figure 4 on and Figure 7 on .

Syntax

```
<complexType name="statisticValues">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:statisticValue[]" />
      </restriction>
   </complexContent>
</complexType>
```

timeInterval

A statistical time interval.

Examples of this complex type being used in a request are given in Figure 3 on page 75 and Figure 5 on page 77.

Syntax

```
<complexType name="timeInterval">
   <sequence>
      <element name="intervalType" nillable="true" type="tns2:timeIntervalType" />
      <element name="length" nillable="true" type="xsd:int" />
      <element name="slideLength" nillable="true" type="xsd:int" />
      <element name="timeProfileName" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

timeProfile

A statistical time profile.

See also "timeIntervalType" on page 52, and the examples of this complex type being used in a request in Figure 9 on page 81.

Syntax

```
<complexType name="timeProfile">
   <sequence>
      <element name="key" nillable="true" type="xsd:string" />
      <element name="intervalType" nillable="true" type="tns2:timeIntervalType" />
      <element name="value" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

timeProfiles

An array of `timeProfile` items.

An example of this complex type being used in a request is given in Figure 9 on .

Syntax

```
<complexType name="timeProfiles">
   <complexContent>
      <restriction base="soapenc:Array">
         <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:timeProfile[]" />
      </restriction>
   </complexContent>
</complexType>
```

timeRangeType

A statistical time range (beginning time and ending time).

Examples of this complex type being used in a request are given in Figure 3 on and Figure 5 on .

Syntax

```
<complexType name="timeRangeType">
   <sequence>
      <element name="leftTime" type="xsd:int" />
      <element name="rightTime" type="xsd:int" />
   </sequence>
</complexType>
```

unsolicitedNotification

If you use `unsolicitedNotification` mode, you must specify the URL of the server used by your client application to receive updates from GIS.

An example of this complex type being used in a request is given in Figure 5 on .

Syntax

```
<complexType name="unsolicitedNotification">
   <sequence>
      <element name="url" nillable="true" type="xsd:string" />
   </sequence>
</complexType>
```

GENESYS
AN ALCATEL-LUCENT COMPANY

Chapter

6

Statistics SDK Service Messages

GIS and your client application communicate using a request and response message flow. Together, each request sent from your client application and the corresponding response message from GIS are called an *operation*.

This chapter examines the messages used for each operation available from the Statistics SDK Service, and then provides diagrams that show the parts and types of each message.

For the syntax used to combine messages into operations, see Chapter 7.

This chapter includes the following sections:

Overview

Messages typically have at least one part, identified by part name and part type. Understanding a message involves understanding the relationships between the message's parts and types.

A part's type can be either a basic data type, such as a string or an integer, a simple type, or a complex type. Simple and complex types are defined in the Schema section of the WSDL file.

If the part type is a complex type, it is made up of a list of one or more types. These subtypes can be simple or complex. To understand and use the message effectively, you must follow the flow from message to part type to subtypes until the data contained in the final type is the basic kind such as a string or an integer.

The sections presenting each message include cross-references that link to the relevant section of Chapter 5.

Message Syntax

This section displays the syntax for both the request and response messages of the following operations:

retrieveStatistic Messages

Request and response messages for the `retrieveStatistic` operation:

- `retrieveStatisticRequest`
- `retrieveStatisticResponse`

See also Figure 3 on page 75 and Figure 4 on page 76.

Syntax

```
<wsdl:message name="retrieveStatisticRequest">
   <wsdl:part name="statistic" type="tns2:statistic"/>
   <wsdl:part name="resource" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="retrieveStatisticResponse">
   <wsdl:part name="retrieveStatisticResult" type="tns2:retrieveStatisticResponse"/>
</wsdl:message>
```

subscribeStatistic Messages

Request and response messages for the `subscribeStatistic` operation:

- `subscribeStatisticRequest`
- `subscribeStatisticResponse`

See also Figure 7 on page 79 and Figure 5 on page 77.

Syntax

```
<wsdl:message name="subscribeStatisticRequest">
   <wsdl:part name="statistic" type="tns2:statistic"/>
   <wsdl:part name="resource" type="xsd:string"/>
```

```
   <wsdl:part name="remoting" type="tns2:unsolicitedNotification"/>
</wsdl:message>

<wsdl:message name="subscribeStatisticResponse"></wsdl:message>
```

retrieveSubscribedStatistics Messages

Request and response messages for the retrieveSubscribedStatistics operation:

- retrieveSubscribedStatisticsRequest
- retrieveSubscribedStatisticsResponse

See also Figure 6 on and Figure 7 on .

Syntax

```
<wsdl:message name="retrieveSubscribedStatisticsRequest">
   <wsdl:part name="subscriptions" type="tns2:statisticSubscriptions"/>
   <wsdl:part name="notification" type="tns2:notification"/>
</wsdl:message>

<wsdl:message name="retrieveSubscribedStatisticsResponse">
   <wsdl:part name="retrieveSubscribedStatisticsResult"
      type="tns2:retrieveSubscribedStatisticsResponse"/>
</wsdl:message>
```

retrieveStatisticalProfile Messages

Request and response messages for the retrieveStatisticalProfile operation:

- retrieveStatisticalProfileRequest
- retrieveStatisticalProfileResponse

See also Figure 8 on and Figure 9 on .

Syntax

```
<wsdl:message name="retrieveStatisticalProfileRequest">
   <wsdl:part name="resource" type="xsd:string"/>
   <wsdl:part name="profileType" type="tns2:statisticalProfileType"/>
</wsdl:message>

<wsdl:message name="retrieveStatisticalProfileResponse">
   <wsdl:part name="retrieveStatisticalProfileResult"
      type="tns2:retrieveStatisticalProfileResponse"/>
</wsdl:message>
```

unsubscribeStatistic Messages

Request and response messages for the unsubscribeStatistic operation:

- unsubscribeStatisticRequest
- unsubscribeStatisticResponse

See also "unsubscribeStatisticRequest" on .

Syntax

```
<wsdl:message name="unsubscribeStatisticRequest">
   <wsdl:part name="statisticId" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="unsubscribeStatisticResponse"></wsdl:message>
```

Message Parts and Types Diagrams

Parts name and define the information that is contained in a message. Each parts consists of a name and a type. To help you analyze the Statistics SDK Service messages, this section contains diagrams of the all relationships between the parts and types for each message.

Diagrams are provided for the following messages:

retrieveStatisticRequest

The message parts and types for the `retrieveStatisticRequest` message is shown in Figure 3.
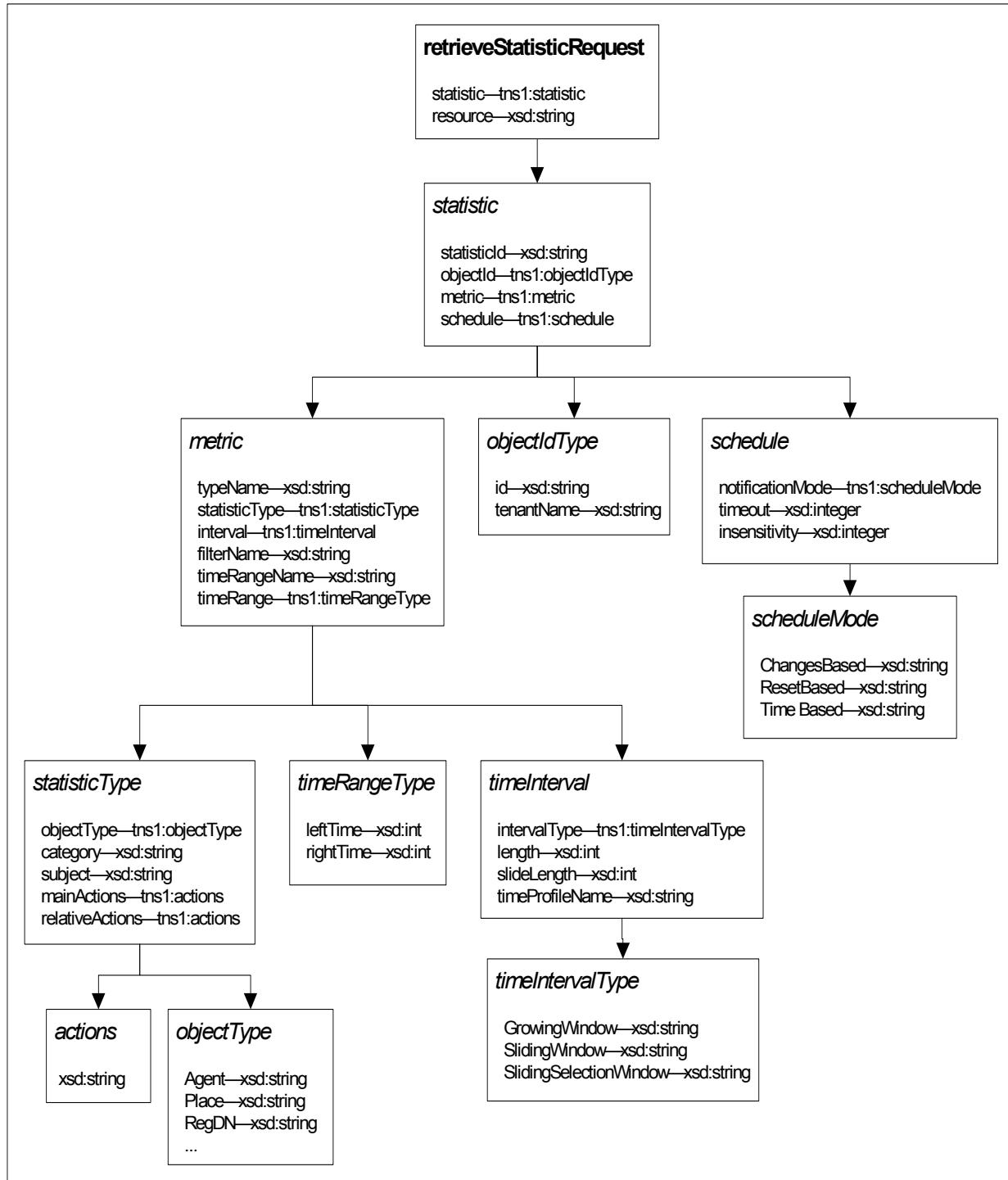
**Figure 3: Message Parts and Types for `retrieveStatisticRequest`**

retrieveStatisticResponse

The message parts and types for the `retrieveStatisticResponse` message is shown in Figure 4.
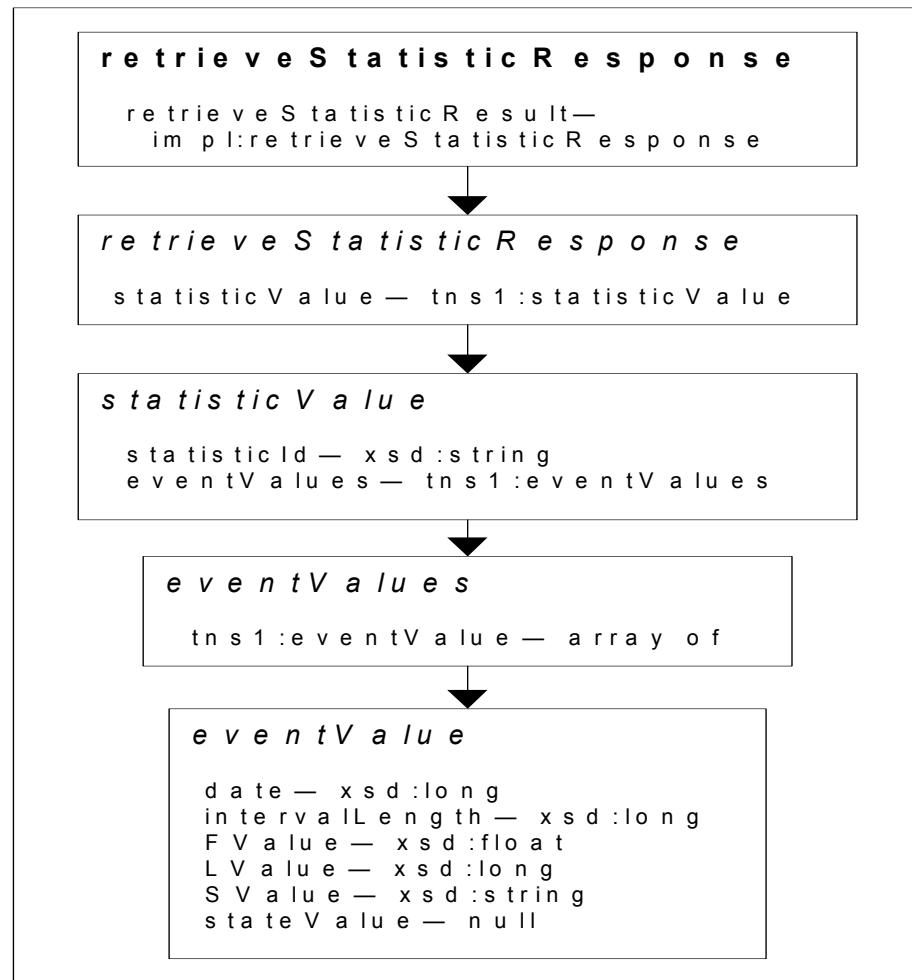
Figure 4: Message Parts and Types for `retrieveStatisticResponse`

subscribeStatisticRequest

The message parts and types for the `subscribeStatisticRequest` message is shown in Figure 5.
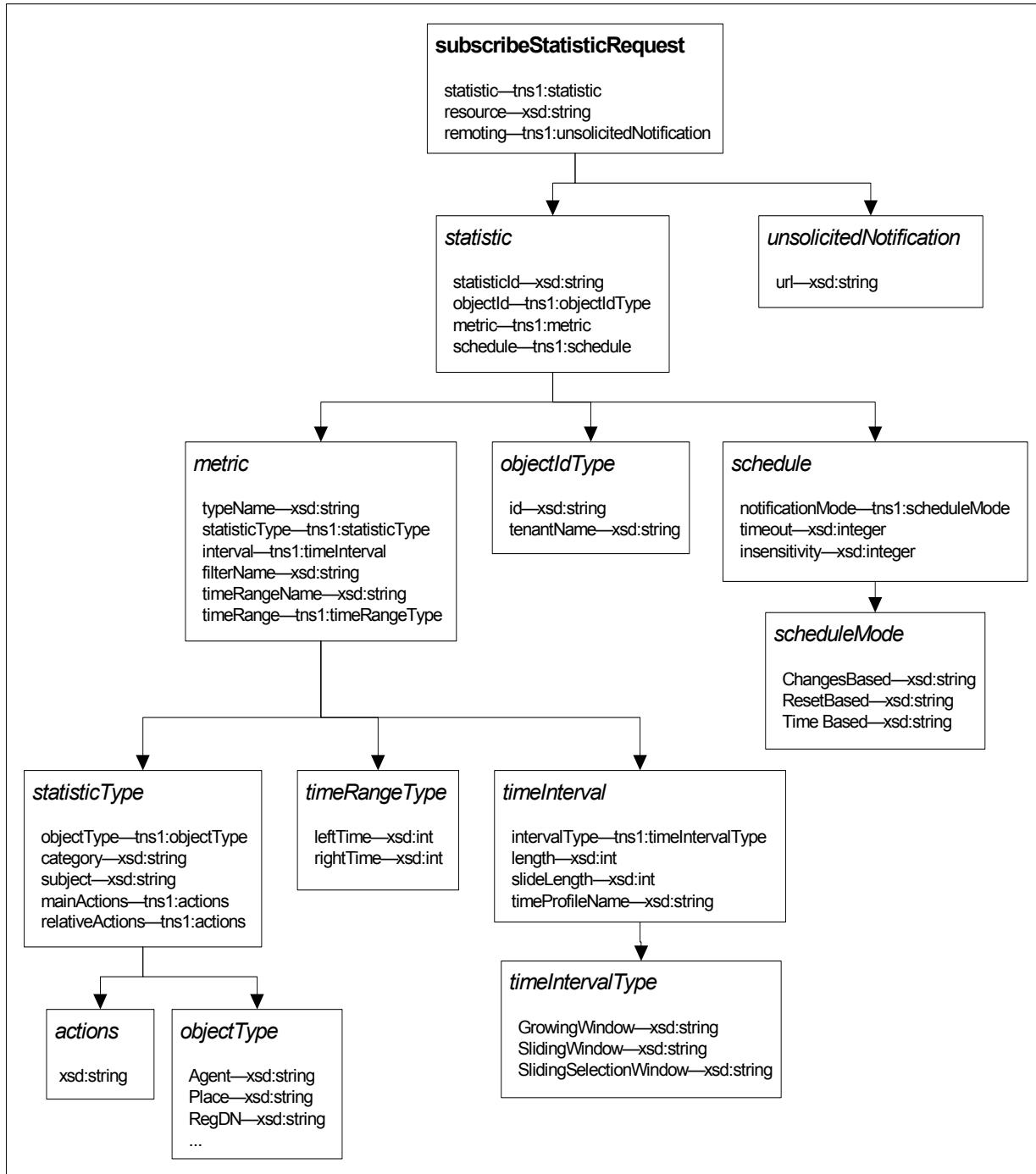
**Figure 5: Message Parts and Types for `subscribeStatisticRequest`**

subscribeStatisticResponse

This message contains no parts.

retrieveSubscribedStatisticsRequest

The message parts and types for the `retrieveSubscribedStatisticsRequest` message is shown in Figure 6.
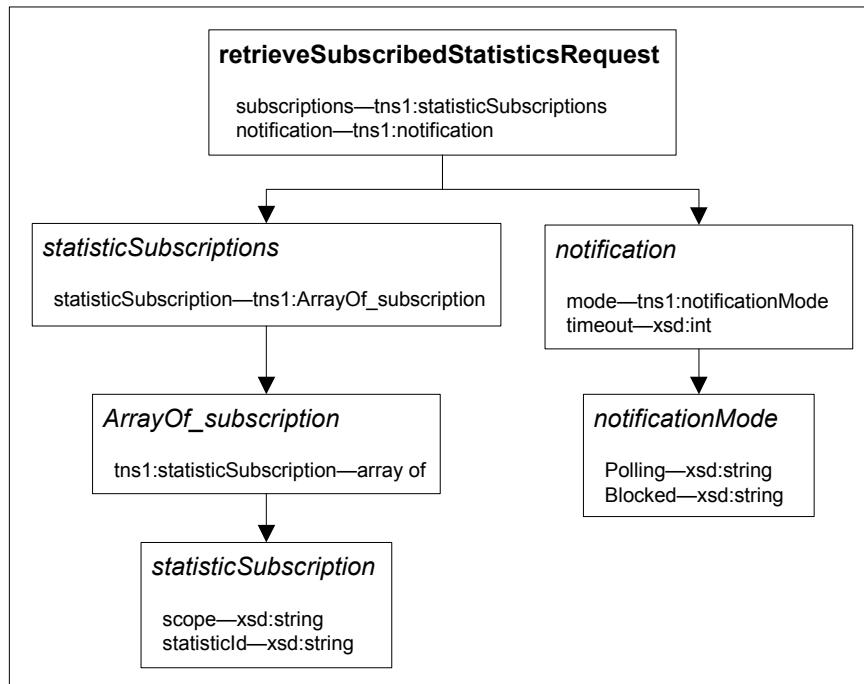
Figure 6: Message Parts and Types for `retrieveSubscribedStatisticsRequest`

retrieveSubscribedStatisticsResponse

The message parts and types for the `retrieveSubscribedStatisticsResponse` message is shown in Figure 7.

```
┌─────────────────────────────────────────────┐
│  ┌───────────────────────────────────────┐  │
│  │ retrieveSubscribedStatisticsResponse  │  │
│  │                                       │  │
│  │ retrieveSubscribedStatisticsResult—   │  │
│  │   impl:retrieveSubscribedStatistics   │  │
│  │   Response                            │  │
│  └───────────────────┬───────────────────┘  │
│                      ▼                       │
│    ┌──────────────────────────────────────┐ │
│    │ retrieveSubscribedStatisticsResponse │ │
│    │                                      │ │
│    │ statisticValues—tns1:statisticValue  │ │
│    └─────────────────┬────────────────────┘ │
│                      ▼                       │
│      ┌─────────────────────────────────┐    │
│      │ statisticValue                  │    │
│      │                                 │    │
│      │ statisticId—xsd:string          │    │
│      │ eventValues—tns1:eventValues    │    │
│      └──────────────┬──────────────────┘    │
│                     ▼                        │
│       ┌──────────────────────────────┐      │
│       │ eventValues                  │      │
│       │                              │      │
│       │ tns1:eventValue—array of     │      │
│       └─────────────┬────────────────┘      │
│                     ▼                        │
│        ┌─────────────────────────────┐      │
│        │ eventValue                  │      │
│        │                             │      │
│        │ date—xsd:long               │      │
│        │ intervalLength—xsd:long     │      │
│        │ FValue—xsd:float            │      │
│        │ LValue—xsd:long             │      │
│        │ SValue—xsd:string           │      │
│        │ stateValue—null             │      │
│        └─────────────────────────────┘      │
└─────────────────────────────────────────────┘
```

Figure 7: Message Parts and Types for `retrieveSubscribedStatisticsResponse`

retrieveStatisticalProfileRequest

The message parts and types for the `retrieveStatisticalProfileRequest` message is shown in Figure 8.
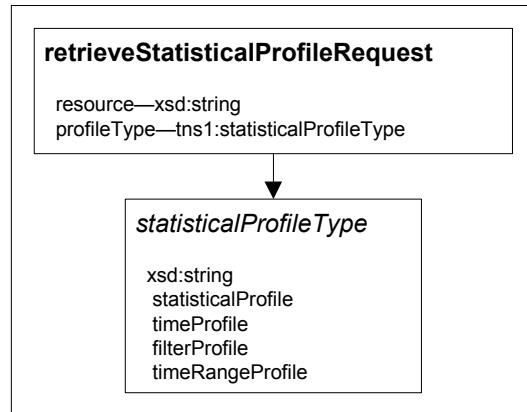
Figure 8: Message Parts and Types for `retrieveStatisticalProfileRequest`

retrieveStatisticalProfileResponse

The message parts and types for the `retrieveStatisticalProfileResponse` message is shown in Figure 9.
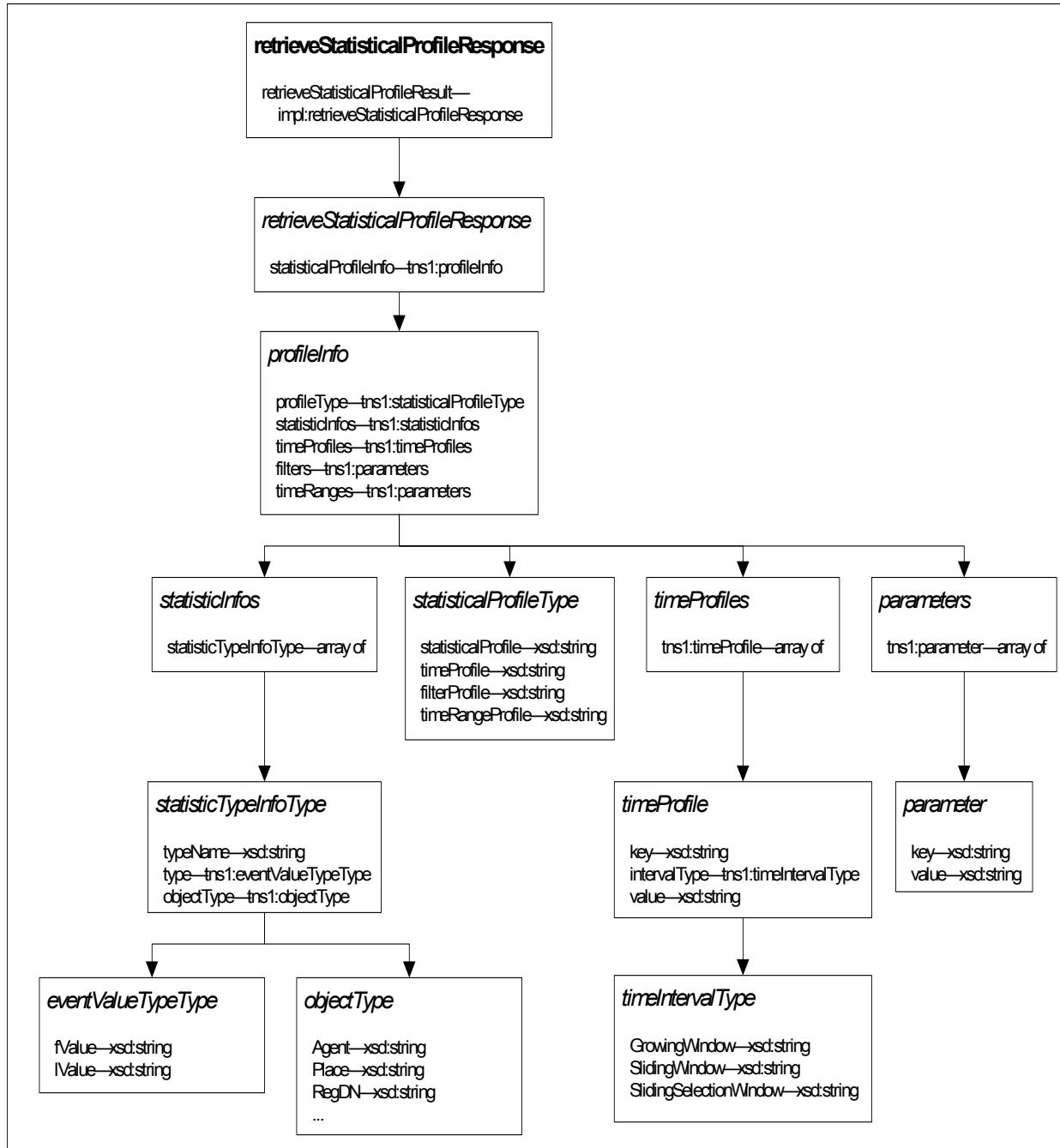
Figure 9: Message Parts and Types for `retrieveStatisticalProfileResponse`

unsubscribeStatisticRequest

The message parts and types for the `unsubscribeStatistic` message is shown in Figure 10.

retrieveStatisticalProfileRequest

resource—xsd:string
profileType—tns1:statisticalProfileType

statisticalProfileType

xsd:string
 statisticalProfile
 timeProfile
 filterProfile
 timeRangeProfile
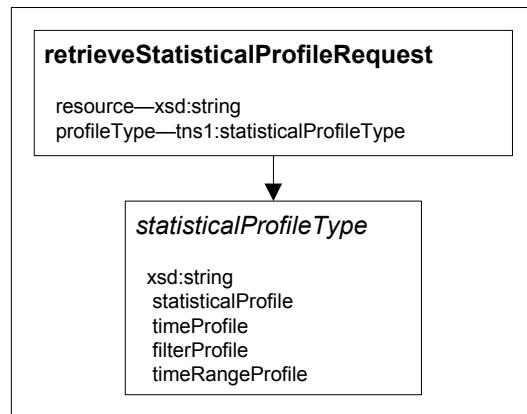
Figure 10: Message Parts and Types for `unsubscribeStatisticRequest`

unsubscribeStatisticResponse

This message includes no parts.

Chapter

7

Statistics SDK Service Operations

This chapter covers the following topics:

Overview

The Statistics SDK Service WSDL file defines five operations, which are taken together to form the Statistics API. As shown in Table 15, the operations fall into two groups: statistical operations and profile operations. Your design for a client application determines how and when these operations are used.

You can retrieve statistical information in two ways:

- Use `retrieveStatistic` to retrieve the current value of a specific statistic.
- Use `subscribeStatistic`, `retrieveSubscribedStatistics`, and `unsubscribeStatistic` to monitor and retrieve statistical information over a period of time.

Note: It is possible to retrieve several subscribed statistics values with one SOAP request. This is done by building a statistic-subscriptions array. For details, see "retrieveSubscribedStatistics Operation" "Example 2: Performance Tip" on page 90.

Table 15: Statistics SDK Service Operations

Operation Group	Operation
Statistical Operations	"retrieveStatistic Operation" on page 84
	"subscribeStatistic Operation" on page 86
	"retrieveSubscribedStatistics Operation" on page 88
	"unsubscribeStatistic Operation" on page 91
Profile Operations	"retrieveStatisticalProfile Operation" on page 93

retrieveStatistic Operation

The `retrieveStatistic` operation retrieves the value of a particular statistic at a single point in time. See "retrieveStatistic Messages" on page 72 to check the syntax required for request and response messages.

Syntax

```
<wsdl:operation name="retrieveStatistic" parameterOrder="statistic resource">
   <wsdl:input message="impl:retrieveStatisticRequest"
     name="retrieveStatisticRequest"/>
   <wsdl:output message="impl:retrieveStatisticResponse"
     name="retrieveStatisticResponse"/>
</wsdl:operation>
```

Example

The following example shows a request using the `retrieveStatistic` operation, and a possible response that is returned for that request.

Request

```
POST /gis/services/StatService?GISsessionId=SessionService:1034175564296H1 HTTP/1.1
Content-Type: text/xml
SOAPAction: ""
User-Agent: Java1.3.1_01
Host: foo-host@location
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-length: 1756
```

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ns0="http://www.genesyslab.com/services/statservice/types/2001/05"
    xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2002/03"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Body>
      <ns0:retrieveStatistic>
         <statistic xsi:type="ns1:statistic">
            <statisticId xsi:type="xsd:string">statref1</statisticId>
            <objectId xsi:type="ns1:objectIdType">
               <id xsi:type="xsd:string">109</id>
               <tenantName xsi:type="xsd:string">Resources</tenantName>
            </objectId>
            <metric xsi:type="ns1:metric">
               <typeName xsi:type="xsd:string">TotalLoginTime</typeName>
               <statisticType xsi:type="ns1:statisticType">
                  <objectType xsi:type="ns1:objectType">Agent</objectType>
               </statisticType>
               <interval xsi:type="ns1:timeInterval">
                  <intervalType xsi:type="ns1:timeIntervalType">GrowingWindow
                     </intervalType>
               </interval>
            </metric>
            <schedule xsi:type="ns1:schedule">
               <notificationMode xsi:type="ns1:scheduleMode">ChangesBased
                  </notificationMode>
            </schedule>
         </statistic>
         <resource xsi:type="xsd:string" xsi:nil="true"/>
      </ns0:retrieveStatistic>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Date: Wed, 09 Oct 2002 15:12:45 GMT
Server: Tomcat Web Server/3.3.1 Final (JSP 1.1; Servlet 2.2)

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
```

```
<ns1:retrieveStatisticResponse
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2001/05">
    <retrieveStatisticResult xsi:type="ns2:retrieveStatisticResponse"
            xmlns:ns2="http://www.genesyslab.com/services/statservice/types/2002/03">
        <statisticValue xsi:type="ns2:statisticValue">
            <statisticId xsi:type="xsd:string" xsi:nil="true"/>
            <eventValues xsi:type="soapenc:Array"
                    soapenc:arrayType="ns2:eventValue[1]"
                    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
                <item>
                    <date xsi:type="xsd:long">1034176681</date>
                    <intervalLength xsi:type="xsd:long">30</intervalLength>
                    <stateValue xsi:type="ns2:statisticState" xsi:nil="true"/>
                    <LValue xsi:type="xsd:long">30</LValue>
                    <SValue xsi:type="xsd:string" xsi:nil="true"/>
                    <FValue xsi:type="xsd:float">0.0</FValue>
                </item>
            </eventValues>
        </statisticValue>
    </retrieveStatisticResult>
</ns1:retrieveStatisticResponse>
</soapenv:Body>
</soapenv:Envelope>
```

subscribeStatistic Operation

The subscribeStatistic operation subscribes to a statistic, which simplifies repeated collection of updated values for this statistic. See "subscribeStatistic Messages" on page 72 to check the syntax required for request and response messages.

Syntax

```
<wsdl:operation name="subscribeStatistic" parameterOrder="statistic resource remoting">
    <wsdl:input message="impl:subscribeStatisticRequest"
        name="subscribeStatisticRequest"/>
    <wsdl:output message="impl:subscribeStatisticResponse"
        name="subscribeStatisticResponse"/>
</wsdl:operation>
```

Example

The following example shows a request using the subscribeStatistic operation, and a possible response that is returned for that request.

Request

```
POST /gis/services/StatService?GISsessionId=SessionService:1034175564296H1 HTTP/1.1
Content-Type: text/xml
SOAPAction: ""
User-Agent: Java1.3.1_01
Host: foo-host@location
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-length: 1848

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
     xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:ns0="http://www.genesyslab.com/services/statservice/types/2001/05"
     xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2002/03"
     SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns0:subscribeStatistic>
       <statistic xsi:type="ns1:statistic">
          <statisticId xsi:type="xsd:string">statref1</statisticId>
          <objectId xsi:type="ns1:objectIdType">
             <id xsi:type="xsd:string">109</id>
             <tenantName xsi:type="xsd:string">Resources</tenantName>
          </objectId>
          <metric xsi:type="ns1:metric">
             <typeName xsi:type="xsd:string">TotalLoginTime</typeName>
             <statisticType xsi:type="ns1:statisticType">
                <objectType xsi:type="ns1:objectType">Agent</objectType>
             </statisticType>
             <interval xsi:type="ns1:timeInterval">
                <intervalType xsi:type="ns1:timeIntervalType">GrowingWindow
                   </intervalType>
             </interval>
          </metric>
          <schedule xsi:type="ns1:schedule">
             <notificationMode xsi:type="ns1:scheduleMode">ChangesBased
                </notificationMode>
          </schedule>
       </statistic>
       <resource xsi:type="xsd:string" xsi:nil="true"/>
       <remoting xsi:type="ns1:unsolicitedNotification" xsi:nil="true"/>
    </ns0:subscribeStatistic>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Date: Wed, 09 Oct 2002 15:06:07 GMT
Server: Tomcat Web Server/3.3.1 Final (JSP 1.1; Servlet 2.2)

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:subscribeStatisticResponse
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2001/05"/>
  </soapenv:Body>
</soapenv:Envelope>
```

retrieveSubscribedStatistics Operation

The `retrieveSubscribedStatistics` operation retrieves the latest value of a subscribed statistic. Use this operation to monitor the values of one or more statistics over time. See "retrieveSubscribedStatistics Messages" on to check the syntax required for request and response messages.

Syntax

```
<wsdl:operation name="retrieveSubscribedStatistics"
    parameterOrder="subscriptions notification">
  <wsdl:input message="impl:retrieveSubscribedStatisticsRequest"
      name="retrieveSubscribedStatisticsRequest"/>
  <wsdl:output message="impl:retrieveSubscribedStatisticsResponse"
      name="retrieveSubscribedStatisticsResponse"/>
</wsdl:operation>
```

Example 1

The following example shows a request using the `retrieveSubscribedStatistics` operation, and a possible response that is returned for that request.

Request

```
POST /gis/services/StatService?GISsessionId=SessionService:1034175564296H1 HTTP/1.1
Content-Type: text/xml
SOAPAction: ""
User-Agent: Java1.3.1_01
```

```
Host: foo-host@location
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-length: 1232

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ns0="http://www.genesyslab.com/services/statservice/types/2001/05"
    xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2002/03"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns0:retrieveSubscribedStatistics>
      <subscriptions xsi:type="ns1:statisticSubscriptions">
        <statisticSubscription xsi:type="ns1:statisticSubscription">
          <scope xsi:type="xsd:string">2</scope>
          <statisticId xsi:type="xsd:string">statref1</statisticId>
        </statisticSubscription>
      </subscriptions>
      <notification xsi:type="ns1:notification">
        <mode xsi:type="ns1:notificationMode">Polling</mode>
        <timeout xsi:type="xsd:integer" xsi:nil="true"/>
      </notification>
    </ns0:retrieveSubscribedStatistics>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Date: Wed, 09 Oct 2002 15:08:42 GMT
Server: Tomcat Web Server/3.3.1 Final (JSP 1.1; Servlet 2.2)

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:retrieveSubscribedStatisticsResponse
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2001/05">
      <retrieveSubscribedStatisticsResult
          xsi:type="ns2:retrieveSubscribedStatisticsResponse"
          xmlns:ns2="http://www.genesyslab.com/services/statservice/types/2002/03">
        <statisticValues xsi:type="soapenc:Array"
            soapenc:arrayType="ns2:statisticValue[1]"
```

```
                        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
                  <item>
                     <statisticId xsi:type="xsd:string">statref1</statisticId>
                     <eventValues xsi:type="soapenc:Array"
                        soapenc:arrayType="ns2:eventValue[2]">
                        <item>
                           <date xsi:type="xsd:long">1034176437</date>
                           <intervalLength xsi:type="xsd:long">154</intervalLength>
                           <stateValue xsi:type="ns2:statisticState" xsi:nil="true"/>
                           <LValue xsi:type="xsd:long">154</LValue>
                           <SValue xsi:type="xsd:string" xsi:nil="true"/>
                           <FValue xsi:type="xsd:float">0.0</FValue>
                        </item>
                        <item>
                           <date xsi:type="xsd:long">1034176435</date>
                           <intervalLength xsi:type="xsd:long">152</intervalLength>
                           <stateValue xsi:type="ns2:statisticState" xsi:nil="true"/>
                           <LValue xsi:type="xsd:long">152</LValue>
                           <SValue xsi:type="xsd:string" xsi:nil="true"/>
                           <FValue xsi:type="xsd:float">0.0</FValue>
                        </item>
                     </eventValues>
                  </item>
               </statisticValues>
            </retrieveSubscribedStatisticsResult>
         </ns1:retrieveSubscribedStatisticsResponse>
      </soapenv:Body>
</soapenv:Envelope>
```

Example 2: Performance Tip

The following example shows a request that optimizes the return of multiple statistics subscriptions using the retrieveSubscribedStatistics operation.

Request

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Date: Wed, 09 Oct 2002 15:08:42 GMT
Server: Tomcat Web Server/3.3.1 Final (JSP 1.1; Servlet 2.2)

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:ns0="http://www.genesyslab.com/services/statservice/types/2001/05"
     xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2002/03"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
```

```
    <ns0:retrieveSubscribedStatistics>
    <subscriptions xsi:type="ns1:statisticSubscriptions">
    <!--  First Subscription -->
       <statisticSubscription xsi:type="ns1:statisticSubscription">
          <scope xsi:type="xsd:string">2</scope>
             <statisticId xsi:type="xsd:string">
                statref1
             </statisticId>
       </statisticSubscription>
    <!--  Second Subscription -->
       <statisticSubscription xsi:type="ns1:statisticSubscription">
          <scope xsi:type="xsd:string">
             2
          </scope>
          <statisticId xsi:type="xsd:string">
             statref2
          </statisticId>
       </statisticSubscription>
    <!-- Third Subscription -->
       <statisticSubscription xsi:type="ns1:statisticSubscription">
          <scope xsi:type="xsd:string">
             2
          </scope>
          <statisticId xsi:type="xsd:string">
             statref3
          </statisticId>
       </statisticSubscription>
    </subscriptions>
    <notification xsi:type="ns1:notification">
       <mode xsi:type="ns1:notificationMode">
          Polling
       </mode>
       <timeout xsi:type="xsd:integer" xsi:nil="true"/>
    </notification>
    </ns0:retrieveSubscribedStatistics>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

unsubscribeStatistic Operation

The unsubscribeStatistic operation unsubscribes from a statistic that you are no longer interested in tracking. See "unsubscribeStatistic Messages" on page 73 to check the syntax required for request and response messages.

Syntax

```
<wsdl:operation name="unsubscribeStatistic" parameterOrder="statisticId">
   <wsdl:input message="impl:unsubscribeStatisticRequest"
```

```
        name="unsubscribeStatisticRequest"/>
    <wsdl:output message="impl:unsubscribeStatisticResponse"
        name="unsubscribeStatisticResponse"/>
</wsdl:operation>
```

Example

The following example shows a request using the `unsubscribeStatistic` operation, and a possible response that is returned for that request.

Request

```
POST /gis/services/StatService?GISsessionId=SessionService:1034175564296H1 HTTP/1.1
Content-Type: text/xml
SOAPAction: ""
User-Agent: Java1.3.1_01
Host: foo-host@location
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-length: 642

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ns0="http://www.genesyslab.com/services/statservice/types/2001/05"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns0:unsubscribeStatistic>
       <statisticId xsi:type="xsd:string">statref1</statisticId>
    </ns0:unsubscribeStatistic>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Date: Wed, 09 Oct 2002 15:10:46 GMT
Server: Tomcat Web Server/3.3.1 Final (JSP 1.1; Servlet 2.2)

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:unsubscribeStatisticResponse
```

```
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2001/05"/>
    </soapenv:Body>
</soapenv:Envelope>
```

retrieveStatisticalProfile Operation

The `retrieveStatisticalProfile` operation retrieves a statistical profile. Statistical profiles define a specific way to collect statistics. Each profile will include a method of calculation, instruction on how to gather the data, and details about how to notify the client. See "retrieveStatisticalProfile Messages" on to check the syntax required for request and response messages.

Syntax

```
<wsdl:operation name="retrieveStatisticalProfile"
      parameterOrder="resource profileType">
  <wsdl:input message="impl:retrieveStatisticalProfileRequest"
      name="retrieveStatisticalProfileRequest"/>
  <wsdl:output message="impl:retrieveStatisticalProfileResponse"
      name="retrieveStatisticalProfileResponse"/>
</wsdl:operation>
```

Example

The following example shows a request using the `retrieveStatisticalProfile` operation, and a possible response that is returned for that request.

Request

```
POST /gis/services/StatService?GISsessionId=SessionService:1034175564296H1 HTTP/1.1
Content-Type: text/xml
SOAPAction: "retrieveStatisticalProfile"
User-Agent: Java1.3.1_01
Host: foo-host@location
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-length: 813

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ns0="http://www.genesyslab.com/services/statservice/types/2001/05"
      xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2002/03"
```

```
        SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
        <ns0:retrieveStatisticalProfile>
            <resource xsi:type="xsd:string" xsi:nil="true"/>
            <profileType xsi:type="ns1:statisticalProfileType">filterProfile</profileType>
        </ns0:retrieveStatisticalProfile>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Date: Wed, 09 Oct 2002 15:01:31 GMT
Server: Tomcat Web Server/3.3.1 Final (JSP 1.1; Servlet 2.2)

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <ns1:retrieveStatisticalProfileResponse
              soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              xmlns:ns1="http://www.genesyslab.com/services/statservice/types/2001/05">
            <retrieveStatisticalProfileResult
                xsi:type="ns2:retrieveStatisticalProfileResponse"
                xmlns:ns2="http://www.genesyslab.com/services/statservice/types/2002/03">
            <statisticalProfileInfo xsi:type="ns2:profileInfo">
            <profileType xsi:type="ns2:statisticalProfileType">filterProfile
                </profileType>
            <statisticInfos xsi:type="soapenc:Array" xsi:nil="true"
                xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"/>
            <timeProfiles xsi:type="soapenc:Array" xsi:nil="true"
                xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"/>
            <filters xsi:type="soapenc:Array" soapenc:arrayType="ns2:parameter[5]"
                xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
            <item>
                <key xsi:type="xsd:string">EMAIL</key>
                <value xsi:type="xsd:string">PairExist(&quot;MediaType&quot;,
                    &quot;E-Mail&quot;)</value>
            </item>
            <item>
                <key xsi:type="xsd:string">CHAT</key>
                <value xsi:type="xsd:string">PairExist(&quot;MediaType&quot;,
                    &quot;Chat&quot;)</value>
            </item>
            <item>
                <key xsi:type="xsd:string">CALL</key>
                <value xsi:type="xsd:string">( ~( PairExist(&quot;MediaType&quot;,
                    &quot;WebCall&quot;) ) ) &amp; ( ~(
```

```
                            PairExist(&quot;MediaType&quot;, &quot;Chat&quot;) ) ) &amp; ( ~(
                            PairExist(&quot;MediaType&quot;, &quot;WebCallIP&quot;) ) )
                            &amp; ( ~( PairExist(&quot;MediaType&quot;,
                            &quot;E-Mail&quot;) ) )</value>
                    </item>
                    <item>
                        <key xsi:type="xsd:string">VOIP</key>
                        <value xsi:type="xsd:string">PairExist(&quot;MediaType&quot;,
                            &quot;WebCallIP&quot;)</value>
                    </item>
                        <item>
                            <key xsi:type="xsd:string">WCALL</key>
                            <value xsi:type="xsd:string">PairExist(&quot;MediaType&quot;,
                                &quot;WebCall&quot;)</value>
                        </item>
                    </filters>
                    <timeRanges xsi:type="soapenc:Array" xsi:nil="true"
                            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"/>
                </statisticalProfileInfo>
            </retrieveStatisticalProfileResult>
        </ns1:retrieveStatisticalProfileResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

![Genesys - An Alcatel-Lucent Company logo]

8 Statistics SDK Service Bindings

Bindings attach a specific protocol (SOAP/HTTP) to each WSDL operation.

This chapter contains the following sections:

Overview

Every SOAP message consists of a SOAP envelope and a SOAP body that is constructed of WSDL-compliant XML.

The SOAP envelope identifies the port, names the message style (RPC), and specifies the transport mechanism (HTTP) that is used for that message. The following example provides an example of how to create a SOAP envelope:

```
<wsdl:binding name="StatServiceSoapBinding" type="impl:StatService">
   <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http">

   SOAP Body

</binding>
```

The SOAP body names the operation, defines the encoding style, and then identifies the GIS namespace for the operation's request and response messages. SOAP body definitions are identical for all of the WSDL operations.

retrieveStatistic Binding

SOAP binding for the `retrieveStatistic` operation.

Syntax

```
<wsdl:operation name="retrieveStatistic">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="retrieveStatisticRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
      </wsdl:input>
   <wsdl:output name="retrieveStatisticResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
      </wsdl:output>
</wsdl:operation>
```

subscribeStatistic Binding

SOAP binding for the `subscribeStatistic` operation.

Syntax

```
<wsdl:operation name="subscribeStatistic">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="subscribeStatisticRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded" />
   </wsdl:input>
   <wsdl:output name="subscribeStatisticResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded" />
   </wsdl:output>
</wsdl:operation>
```

retrieveSubscribedStatistics Binding

SOAP binding for the `retrieveSubscribedStatistics` operation.

Syntax

```
<wsdl:operation name="retrieveSubscribedStatistics">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="retrieveSubscribedStatisticsRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="retrieveSubscribedStatisticsResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

unsubscribeStatistic Binding

SOAP binding for the `unsubscribeStatistic` operation.

Syntax

```
<wsdl:operation name="unsubscribeStatistic">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="unsubscribeStatisticRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="unsubscribeStatisticResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

retrieveStatisticalProfile Binding

SOAP binding for the `retrieveStatisticalProfile` operation.

Syntax

```
<wsdl:operation name="retrieveStatisticalProfile">
   <wsdlsoap:operation soapAction=""/>
   <wsdl:input name="retrieveStatisticalProfileRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:input>
   <wsdl:output name="retrieveStatisticalProfileResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.genesyslab.com/services/statservice/wsdl/2002/03"
         use="encoded"/>
   </wsdl:output>
</wsdl:operation>
```

Index

A

B

C

Configuration Data

D

E

F

G

I

L

M

N