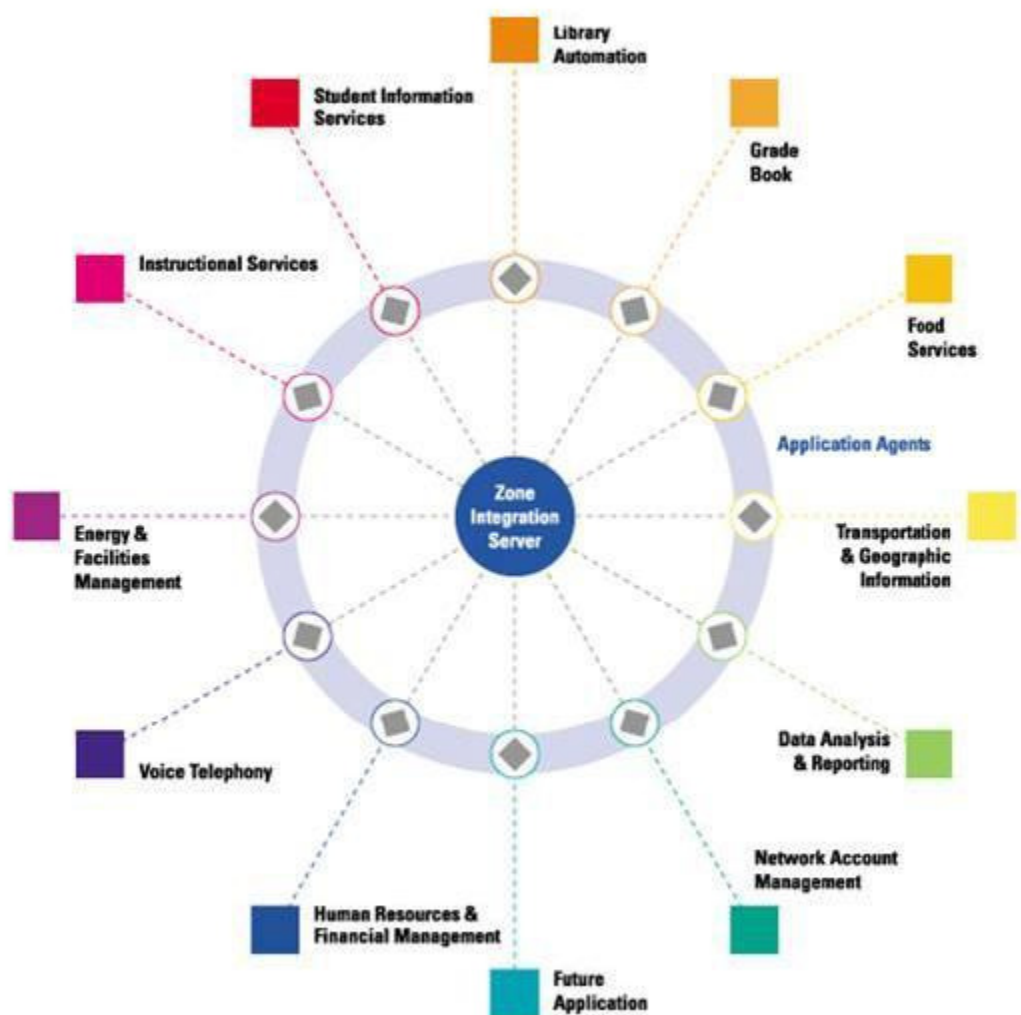


Schools Interoperability Framework™

SIF Infrastructure Implementation Specification 2.5

February 24, 2011



This version:

<http://specification.sifassociation.org/Implementation/2.5/Infrastructure/3/>

Previous version:

<http://specification.sifassociation.org/Implementation/2.4/>

Latest version:

<http://specification.sifassociation.org/Implementation/>

Schemas

[SIF_Message](#) (single file, non-annotated) ([ZIP archive](#))

[SIF_Message](#) (single file, annotated) ([ZIP archive](#))

[SIF_Message](#) (includes, non-annotated) ([ZIP archive](#))

[SIF_Message](#) (includes, annotated) ([ZIP archive](#))

Note: [SIF_Message](#) schemas define every data object element as optional per SIF's Publish/Subscribe and SIF Request/Response Models; [DataModel](#) schemas maintain the cardinality of all data object elements.

Please refer to the [errata](#) for this document, which may include some normative corrections.

This document is also available in these non-normative formats: [ZIP archive](#), [PDF](#) (for printing as a single file), [Excel spreadsheet](#).

Copyright ©2011 [Schools Interoperability Framework \(SIF™\) Association](#). All Rights Reserved.

1 Preamble

1.1 Abstract

1.1.1 What is SIF?

The Schools Interoperability Framework (SIF) is not a product, but a technical blueprint for enabling diverse applications to interact and share data related to entities in the pK-12 instructional and administrative environment. SIF is designed to:

- Facilitate data sharing and reporting between applications without incurring expensive customer development costs;
- Enhance product functionality efficiently; and
- Provide best-of-breed solutions to customers easily and seamlessly.

The SIF Implementation Specification defines:

- an XML-based messaging framework that allows diverse software applications to interoperate and share and report data related to entities in the pK-12 instructional and administrative environment;
- an HTTP(S)-based transport for conveying these SIF messages;
- an abstract, platform-independent definition of a message queue for reliable delivery of asynchronous SIF messages and related synchronous administrative functions—the *Zone Integration Server (ZIS)*; and
- an abstract, platform-independent definition of the interface between a software application and the ZIS—the *SIF Agent*.

These are known collectively as the *SIF Infrastructure*. The SIF Implementation Specification also defines the *SIF Data Model*:

- an XML-based data model that models entities in the pK-12 environment as *SIF Data Objects* to be shared between applications.

A *SIF Zone* is a distributed system that consists of a ZIS and one or more software applications with a SIF Agent (a *SIF-enabled application*) sharing/reporting one or more SIF data objects over a network. A *SIF Implementation* consists of one or more SIF Zones deployed and configured to meet customer data sharing and reporting needs.

The SIF Implementation Specification defines architecture requirements and communication protocols for software components and the interfaces between them; it makes no assumption of specific hardware or software products needed to develop SIF-enabled applications and Zone Integration Server implementations, other than their ability to support technologies leveraged as the foundation for SIF, most prominently XML and HTTP(S).

1.1.2 Schools Interoperability Framework Association

The Schools Interoperability Framework Association (SIF Association) is an industry initiative to enable interoperability and data sharing between software applications in the pK-12 instructional and administrative environment, and the forum for companies and educators to participate in the development of SIF specifications in the SIF Association's working groups and task forces. The SIF Association is designed to:

- Join industry leaders in creating the next-generation framework for education technology; and
- Leverage co-marketing opportunities with partners and distributors.

1.2 Disclaimer

The information, software, products, and services included in the SIF Implementation Specification may include inaccuracies or typographical errors. Changes are periodically added to the information herein. The SIF Association may make improvements and/or changes in this document at any time without notification. Information contained in this document should not be relied upon for personal, medical, legal, or financial decisions. Appropriate professionals should be consulted for advice tailored to specific situations.

THE SIF ASSOCIATION, ITS PARTICIPANT(S), AND THIRD PARTY CONTENT PROVIDERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY, RELIABILITY, TIMELINESS, AND ACCURACY OF THE INFORMATION, SOFTWARE, PRODUCTS, SERVICES, AND RELATED GRAPHICS CONTAINED IN THIS DOCUMENT FOR ANY PURPOSE. ALL SUCH INFORMATION, SOFTWARE, PRODUCTS, SERVICES, AND RELATED GRAPHICS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE SIF ASSOCIATION AND/OR ITS PARTICIPANT(S) HEREBY DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS INFORMATION, SOFTWARE, PRODUCTS, SERVICES, AND RELATED GRAPHICS, INCLUDING ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT.

IN NO EVENT SHALL THE SIF ASSOCIATION, ITS PARTICIPANT(S), OR THIRD PARTY CONTENT PROVIDERS BE LIABLE FOR ANY DIRECT, INDIRECT, PUNITIVE, INCIDENTAL, SPECIAL, CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF USE, DATA, OR PROFITS, ARISING OUT OF OR IN ANY WAY CONNECTED WITH THE USE OR PERFORMANCE OF THIS DOCUMENT, WITH THE DELAY OR INABILITY TO USE THE DOCUMENT, THE PROVISION OF OR FAILURE TO PROVIDE SERVICES, OR FOR ANY INFORMATION, SOFTWARE, PRODUCTS, SERVICES AND RELATED GRAPHICS OBTAINED THROUGH THIS DOCUMENT OR OTHERWISE ARISING OUT OF THE USE OF THIS DOCUMENT, WHETHER BASED ON CONTRACT, TORT, STRICT LIABILITY, OR OTHERWISE, EVEN IF THE SIF ASSOCIATION, ITS PARTICIPANT(S), OR THIRD PARTY CONTENT PROVIDERS HAVE BEEN ADVISED OF THE POSSIBILITY OF DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. IF YOU ARE DISSATISFIED WITH ANY PORTION OF THIS DOCUMENT OR WITH ANY OF THESE TERMS OF USE, YOUR SOLE AND EXCLUSIVE REMEDY IS TO DISCONTINUE USING THIS DOCUMENT.

This specification is released with the following provisos to developers and educators.

1.3 Certification and Compliance Claims

Though a product may be demonstrated to comply with this specification, no product may be designated as *SIF Certified™* by an organization or individual until the product has been tested against and passed established compliance criteria, published separately

[[SIFCertification](#)]. Organizations and individuals that are currently paying annual membership dues to the SIF Association and dedicating resources to the initiative may also use the designation *SIF Participant* to describe their involvement with the SIF Association and SIF in marketing, public relations and other materials.

Quick Table of Contents

1

Preamble**2 Introduction****3 Architecture****4 Messaging****5 Infrastructure****A Common Types****B Code Sets****C Notes on Related Technologies****D Wildcard Version Support Implementation Notes****E Selective Message Blocking (SMB) Example****F Index of Tables****G Index of Examples****H Index of Figures****I Index of Objects****J Index of Common Elements****K Index of Common Types****L Index of Elements****M Index of Attributes****N References**

Full Table of Contents

1 Preamble

1.1 Abstract

1.1.1 What is SIF?

1.1.2 Schools Interoperability Framework Association

1.2 Disclaimer

1.3 Certification and Compliance Claims

2 Introduction

2.1 Specification Organization

2.2 Document Conventions

2.2.1 Definitions

2.2.2 Structure and Values

2.2.3 Examples

2.2.4 References

2.2.5 Terminology

2.2.6 XML Diagrams

2.3 Version Numbers

2.4 Highlighted Additions/Changes Since Version 2.4

2.4.1 New Web Services Transport

2.4.2 Directed Events

2.4.3 Stand-Alone Transports

2.4.4 Infrastructure Documentation

3 Architecture

3.1 Assumptions

3.1.1 Notes on Related Technologies

3.2 Architectural Components

3.2.1 SIF Systems

3.3 Concepts

3.3.1 Data Model

3.3.2 Zone Architecture

3.3.2.1 Contexts

3.3.3 Infrastructure and Messaging

3.3.4 Data Provision: A Request/Response Model

3.3.4.1 Routing

3.3.4.2 Access Control

3.3.4.3 Error Reporting

3.3.5 Event Reporting: A Publish/Subscribe Model

3.3.6 Communication: An Asynchronous Model

3.3.6.1 Guaranteed Delivery

3.3.7 Security Model

3.3.7.1 Encryption

3.3.7.2 Authentication and Validation

3.3.7.3 Access Control

3.3.8 Zone Services

3.3.9 Naming Conventions for Agents and Zone Integration Servers

3.3.10 Object Identifiers

3.3.10.1 Object Identifier Persistence

3.4 Agent/Application Requirements

3.4.1 Communicate with the ZIS

3.4.1.1 HTTPS

3.4.1.2 Agent Registration

3.4.2 Transmit Application Changes to the ZIS

3.4.3 Respond to Requests

3.4.4 Vendor Application Support for SIF

3.4.5 Support Authentication and Digital Signatures

3.4.6 Agent Local Queue

3.4.7 Wildcard Version Support

3.5 Zone Integration Server Requirements

3.5.1 Access Control List

3.5.2 Zone Status

3.5.3 SIF XML Filter

3.5.3.1 SIF XML Filter Process Rules

3.5.3.2 Implementation of SIF XML Filter Syntax

3.5.4 Zone Context Registry

3.5.5 Administration

3.5.6 Support Selective Message Blocking (SMB) to Resolve Deadlocks

3.5.6.1 Description

3.5.6.2 Requirements

3.5.6.3 Example

3.5.7 Quality of Service Implementation

3.6 Message Processing

3.6.1 Message Validation

3.6.2 Message Identification

3.6.3 Message Security

3.6.3.1 SIF_AuthenticationLevel

3.6.3.2 SIF_EncryptionLevel

3.6.3.3 Notes on SIF_AuthenticationLevel

3.6.3.4 Notes on SIF_EncryptionLevel

3.6.4 Message Robustness

3.6.5 Message Cycle

3.6.6 Message Delivery

3.6.6.1 The "Push" Model

3.6.6.2 The "Pull" Model

3.6.6.2.1 Version Management

3.6.6.3 "Multiple Version" Zones

3.7 Infrastructure Transport Layer

3.7.1 SIF HTTPS Transport

3.7.1.1 HTTPS Request/Response Model

3.7.1.2 UTF-8 Encoding

3.7.1.3 HTTP Request Headers

3.7.1.4 HTTP Response Headers

3.7.1.5 100 (Continue)

3.7.1.6 3XX, 4XX, 5XX Notices

3.7.2 SIF HTTP Transport

3.7.3 SIF HTTP(S) Transport Compression

3.7.4 SIF_Protocol/SIF_Property Accept-Encoding

3.7.5 HTTP Client Requirements

3.7.6 HTTP Server Requirements

3.7.7 Push-Mode Agent Requirements

3.7.8 Zone Integration Server Transport Requirements

4 Messaging

4.1 Agent Protocols

4.1.1 Agent Messaging Protocols

4.1.1.1 SIF_Register

4.1.1.2 SIF_Unregister

4.1.1.3 SIF_Provide

4.1.1.4 SIF_Unprovide

4.1.1.5 SIF_Subscribe

4.1.1.6 SIF_Unsubscribe

4.1.1.7 SIF_Provision

4.1.1.8 SIF_Event

4.1.1.9 SIF_Request

4.1.1.10 SIF_Ping

4.1.1.11 SIF_Sleep

4.1.1.12 SIF_Wakeup

4.1.1.13 SIF_GetZoneStatus

4.1.1.14 SIF_GetAgentACL

4.1.1.15 SIF_CancelRequests

4.1.1.16 SIF_GetMessage (Pull-Mode only)

4.1.1.17 SIF_Ack (Push-Mode)

4.1.1.18 SIF_Ack (Pull-Mode)

4.1.1.19 SIF_ServiceNotify

4.1.1.20 SIF_ServiceInput

4.1.1.21 SIF_CancelServiceInputs

4.1.2 Agent Message Handling Protocols

4.1.2.1 SIF_Message

- 4.1.2.2 SIF_Event
- 4.1.2.3 SIF_Request
- 4.1.2.4 SIF_Response
- 4.1.2.5 SIF_Ping (Push-Mode only)
- 4.1.2.6 SIF_Sleep (Push-Mode only)
- 4.1.2.7 SIF_Wakeup (Push-Mode only)
- 4.1.2.8 SIF_CancelRequests (Push-Mode only) (optional)
- 4.1.2.9 SIF_CancelServiceInputs (Push-Mode only) (optional)
- 4.1.2.10 SIF_ServiceNotify
- 4.1.2.11 SIF_ServiceInput
- 4.1.2.12 SIF_ServiceOutput

4.2 ZIS Protocols

4.2.1 ZIS Messaging Protocols

- 4.2.1.1 SIF_Message Delivery (SIF_Event, SIF_Request, SIF_Response, SIF_ServiceInput, SIF_ServiceOutput, SIF_ServiceNotify to a Push-mode Agent)
- 4.2.1.2 SIF_Ping (to a Push-mode Agent)
- 4.2.1.3 SIF_Sleep (to a Push-mode Agent)
- 4.2.1.4 SIF_Wakeup (to a Push-mode Agent)
- 4.2.1.5 SIF_CancelRequests (to a Push-mode Agent)
- 4.2.1.6 SIF_CancelServiceInputs (to a Push-mode Agent)

4.2.2 ZIS Message Handling Protocols

- 4.2.2.1 SIF_Message
- 4.2.2.2 SIF_Register
- 4.2.2.3 SIF_Unregister
- 4.2.2.4 SIF_Provide
- 4.2.2.5 SIF_Unprovide
- 4.2.2.6 SIF_Subscribe
- 4.2.2.7 SIF_Unsubscribe
- 4.2.2.8 SIF_Provision
- 4.2.2.9 SIF_Event
- 4.2.2.10 SIF_Request
- 4.2.2.11 SIF_Response
- 4.2.2.12 SIF_Ping
- 4.2.2.13 SIF_Sleep
- 4.2.2.14 SIF_Wakeup
- 4.2.2.15 SIF_GetZoneStatus
- 4.2.2.16 SIF_GetAgentACL
- 4.2.2.17 SIF_CancelRequests
- 4.2.2.18 SIF_CancelServiceInputs
- 4.2.2.19 SIF_GetMessage
- 4.2.2.20 SIF_Ack (from a Push-Mode Agent)
- 4.2.2.21 SIF_Ack (from a Pull-Mode Agent)
- 4.2.2.22 SIF_ServiceNotify
- 4.2.2.23 SIF_ServiceInput
- 4.2.2.24 SIF_ServiceOutput

5 Infrastructure

5.1 Common Elements

- 5.1.1 SIF_ExtendedElements
- 5.1.2 SIF_Message

5.1.3 SIF_Header**5.1.4 SIF_EncryptionLevel****5.1.5 SIF_AuthenticationLevel****5.1.6 SIF_Contexts****5.1.7 SIF_Context****5.1.8 SIF_Protocol****5.1.9 SIF_Status****5.1.10 SIF_Error****5.1.11 SIF_Query****5.1.11.1 SIF_ConditionGroup****5.1.11.2 SIF_Element Syntax****5.1.12 SIF_ExtendedQuery****5.1.12.1 Mapping SIF_Query to SIF_ExtendedQuery****5.1.13 SIF_ExtendedQueryResults****5.2 Messages****5.2.1 SIF_Ack****5.2.2 SIF_Event****5.2.3 SIF_Provide****5.2.4 SIF_Provision****5.2.5 SIF_Register****5.2.6 SIF_Request****5.2.7 SIF_Response****5.2.8 SIF_Subscribe****5.2.9 SIF_SystemControl****5.2.10 SIF_Ping****5.2.11 SIF_Sleep****5.2.12 SIF_Wakeup****5.2.12.1 SIF_Sleep/SIF_Wakeup versus SIF_Register/SIF_Unregister****5.2.13 SIF_GetMessage****5.2.14 SIF_GetZoneStatus****5.2.15 SIF_GetAgentACL****5.2.16 SIF_CancelRequests****5.2.17 SIF_CancelServiceInputs****5.2.18 SIF_Unprovide****5.2.19 SIF_Unregister****5.2.20 SIF_Unsubscribe****5.2.21 SIF_ServiceInput****5.2.22 SIF_ServiceOutput****5.2.23 SIF_ServiceNotify****5.3 Objects****5.3.1 SIF_AgentACL**

5.3.2 SIF_LogEntry

5.3.3 SIF_ZoneStatus

A Common Types

A.1 AbstractContentType

A.2 AbstractPackageType

A.3 DefinedProtocolsType

A.4 ExtendedContentType

A.5 GUIDType

A.6 IdRefType

A.7 MsgIdType

A.8 ObjectNameType

A.9 ObjectType

A.10 RefIdType

A.11 ReportDataObjectType

A.12 ReportPackageType

A.13 SelectedContentType

A.14 ServiceNameType

A.15 URIOrBinaryType

A.16 VersionType

A.17 VersionWithWildcardsType

B Code Sets

Infrastructure

Status Code

Error Category

XML Validation Error

Encryption Error

Authentication Error

Access and Permission Error

Registration Error

Provision Error

Subscription Error

Request and Response Error

Event Reporting and Processing Error

Transport Error

System Error

Generic Message Handling Error

SMB Error

SIF Zone Service Error

SIF_LogEntry

Agent Error Condition

Data Issues with Failure Result

Data Issues with Success Result

Success Category

ZIS Error Condition

C Notes on Related Technologies

C.1 SIF and HTTP(S)

C.2 SIF and URLs

C.3 SIF and XML

C.4 SIF and Unicode

C.5 SIF and XPath

C.6 SIF and XML Schema

C.6.1 xs:boolean

C.6.2 xs:time

C.6.3 xs:date

C.6.4 xs:dateTime

C.7 SIF and XML Namespaces

C.8 SIF and UUIDs/GUIDs

C.9 SIF and Web Services

D Wildcard Version Support Implementation Notes

D.1 XML Parsing

D.2 XML Validation

D.3 SIF_Message Handling

E Selective Message Blocking (SMB) Example

E.1 Example

F Index of Tables

G Index of Examples

H Index of Figures

I Index of Objects

J Index of Common Elements

K Index of Common Types

L Index of Elements

M Index of Attributes

N References

2 Introduction

2.1 Specification Organization

This document, the *SIF Infrastructure Implementation Specification*, contains the SIF Infrastructure. This includes

[Architecture](#), [Messaging](#), and [Infrastructure](#) sections. A separate document, the *SIF Data Model Implementation Specification*, contains the SIF Data Model. See Chapter 1, Preamble, for an explanation of Infrastructure and Data Model. This document will be of interest to technical readers, including software architects, developers and integrators.

The [Preamble](#) provides an abstract of SIF along with the SIF Association disclaimer and details regarding certification and compliance claims.

This Introduction outlines the organization of the specification, provides conventions used in this document, and summarizes versioning of the specification. Highlights of additions/changes since the previous version of the specification are also provided.

[Architecture](#) describes the assumptions, concepts, models, and requirements related to the SIF infrastructure and data model.

[Messaging](#) details the actions Agents and Zone Integration Servers take when sending and receiving messages.

[Infrastructure](#) provides definitions of the XML structure of elements, messages and objects related to SIF infrastructure as opposed to data in the pK-12 environment.

The document concludes with various appendices including lists of code set values defined within SIF and in external documents, and ends with a list of [references](#) to other documents.

2.2 Document Conventions

2.2.1 Definitions

The first time a term or concept is defined, it may be *emphasized*.

2.2.2 Structure and Values

SIF message and object names, XML element tags, attribute names and values, and other codes or values are typically presented as in this sentence.

2.2.3 Examples

Longer examples of XML or HTTP messages are typically numbered and presented as given here.

Example 2.2.3-1: Examples Convention

2.2.4 References

References to other works occurring in this text are given in brackets, e.g. [REFERENCE]. The text in brackets corresponds to a key in the

[References](#) appendix. Often when the text in the brackets duplicates surrounding text, the reference alone is used (e.g. [XML] instead of XML [XML]).

2.2.5 Terminology

The key words [MUST](#), [MUST NOT](#), [REQUIRED](#), [SHALL](#), [SHALL NOT](#), [SHOULD](#), [SHOULD NOT](#), [RECOMMENDED](#), [MAY](#), [OPTIONAL](#), when [EMPHASIZED](#), are to be interpreted as described in

[RFC 2119].

2.2.6 XML Diagrams

Quick overviews of XML structures, including messages, objects, common elements and types, are provided in XML diagrams. The following diagram illustrates the conventions typically encountered in SIF.

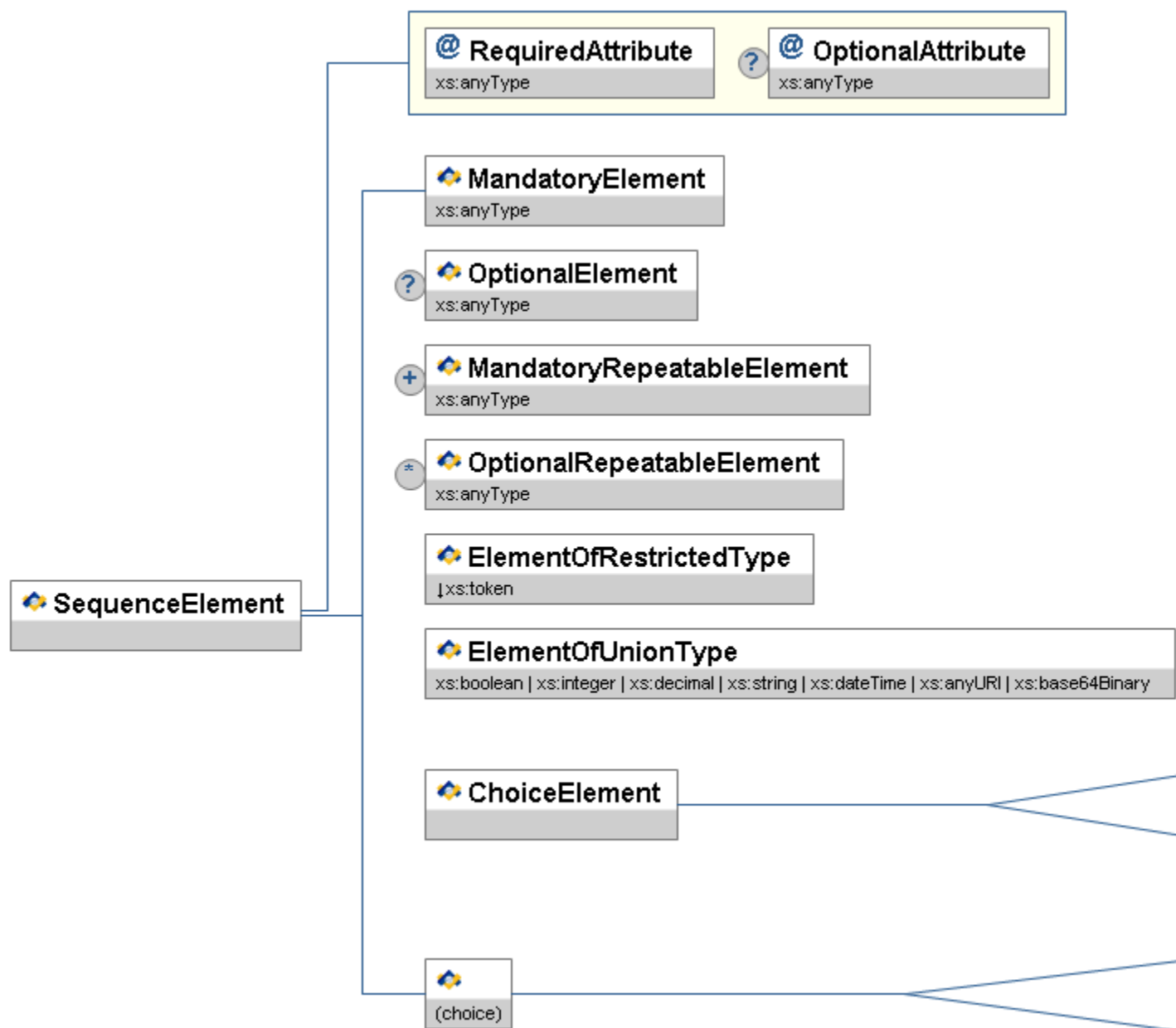


Figure 2.2.6-1: XML Diagram Conventions

XML elements are represented by rectangles with the name of the element in the upper portion and the type, if any, in the lower portion. Attributes are represented in the same fashion, but have an @ icon rather than a SIF icon. Elements and attributes that are optional have a circled ? (0 or 1 occurrence) to the left of the rectangle. Optional and mandatory repeatable elements are indicated by a circled * (0 or more occurrences) and + (1 or more occurrences), respectively. Element attributes are grouped together in a rectangular block and connected to the element with a line that turns at right angles. Ordered sequences of XML elements are bracketed by lines that turn at right angles. When a choice of XML elements is indicated, the elements are bracketed by angled lines. A choice of elements can occur within an element, or may be an unnamed choice of elements.

XML types are represented using the same conventions as for XML elements, though the type portion of the rectangle typically indicates a base type, if any.

The type name of any element, attribute or type may be prefixed with a `|`, indicating the type is restricted in some fashion by one or more XML Schema facets (e.g. enumeration). When the type is a union of types, a list of types is presented, each type separated by `|`; if the list of union types is long, the list may be marked with ellipses, e.g., `| ...`.

In an actual XML diagram, element, type and attribute rectangles are usually linked to their corresponding definitions/descriptions in accompanying tables.

2.3 Version Numbers

The SIF Implementation Specification uses the following version numbering scheme:

major version.minor version`±`*revision number*

Major versions typically introduce additions/changes to the SIF infrastructure and/or data model changes that impact a significant percentage of SIF-enabled applications (e.g. making previously optional elements mandatory, removal of deprecated objects, elements or values). The first release of a major version has a minor version of `0` (`2.0`); major version numbers start at `1` and are incremented as major versions are released (`1.0`, `2.0`, `3.0`, ...).

Minor releases typically introduce new data objects, or optional additions to data objects, to the marketplace, and may include minor infrastructure additions/changes that do not impact existing SIF-enabled applications and that ZIS vendors have agreed to implement. The first minor version released subsequent to and within a major release has a minor version of `1` and is incremented as new minor versions are released (`2.1`, `2.2`, ...). If a significant number of minor release features is introduced in a specification, the SIF Association may decide to increment the minor version number by more than `1` (e.g. `1.1` to `1.5`), though a number like `1.5` is not an indication of being halfway to a major release, as minor version numbers may be incremented significantly past `10` (`2.10`, `2.11`, ...) as data objects and other minor version features are released.

Corrections resulting from identified errata, as well as textual changes, may be incorporated into a revision release. These typically include minor corrections to messages or data objects, corrections of typographical errors, or corrected/expanded documentation. If major errors in any release are identified, a revision release may incorporate changes more typical of a major or minor release. First major and minor releases have a revision number of `0`, which is omitted from the version number (`2.0`, not `2.0r0`); subsequent revision numbers start at `1` and are incremented as new revisions are released (`2.0r1`, `2.0r2`, ...).

2.4 Highlighted Additions/Changes Since Version 2.4

This release contains the following significant updates and extensions to the SIF Infrastructure Implementation Specification.

2.4.1 New Web Services Transport

The new Web Services (SOAP) transport is defined in an appendix to the v.2.5 Infrastructure document. For SIF v2.5, this appendix section is self-contained and completely specifies how to use SOAP and WSDL to transport SIF data objects. In a future release, this section may become more integrated with the rest of the Infrastructure document.

2.4.2 Directed Events

An optional feature related to Event and Service Notify messages has been added. In past releases, the ZIS would ignore the Destination Id element and distribute the Event to all subscribed agents in the zone. Beginning with this release (v. 2.5) the ZIS will route the Event (or Service Notify) message only to the agent specified in the Destination Id element, if the element is present. The handling of Destination Id information is described in section 5.1.3 of the SIF Infrastructure Implementation Specification.

2.4.3 Stand-Alone Transports

As a result of the work accomplished in version 2.5, we now have two stand-alone transports that can be used with any existing or future data model, namely:

1. The Web Services transport, and
2. The traditional HTTP/HTTPS transport.

2.4.4 Infrastructure Documentation

The following documentation changes have been made to the Infrastructure chapters of the Specification. No changes to the meaning or intent of the Specification were made as a result of these edits:

Chapters 3 and 5 have been edited for clarity and Standard English language usage.

Flow Charts have been added to Infrastructure Chapter 4 to help clarify the messaging protocols. A note has been included that if the text and the chart conflict, the text always takes precedence.

Orphan (never used) types have been removed.

Some paragraphs or small sections have been rearranged or brought together to reduce confusion or conflicting statements on a topic.

3 Architecture

This section describes the architecture and components that make up SIF. It presents high-level functional requirements for each component and interfaces between them. More detail on particular requirements and interfaces may be found in

[Messaging](#) and [Infrastructure](#).

3.1 Assumptions

The following assumptions are made of non-technical readers of this specification, especially end users undertaking SIF implementations:

- A passing familiarity with

- [\[XML\]](#) and its use. Readers lacking this prerequisite are referred to [\[XMLINTRO\]](#) and other ubiquitous materials.

- A familiarity with HTTP and HTTPS and the security, encryption and authentication features of the latter. These should be familiar to World Wide Web users.

- A good understanding of the educational data that applications in an implementation store and would benefit from sharing, and the ability to identify equivalents in the SIF Data Model.

They should also be aware that there are numerous third-party products and services available to aid in SIF implementation and integration.

Technical readers implementing SIF software and software solutions, particularly those implementing SIF agents and Zone Integration Servers from scratch as opposed to using or building upon third-party products and services, should have an understanding of:

- The subset of HTTP 1.1 [\[RFC 2616\]](#) referenced in the SIF HTTPS and SIF HTTP Transport Layers.

- Those portions of TLS 1.0 [\[RFC 2246\]](#), SSL 3.0 [\[SSL3\]](#) and SSL 2.0 [\[SSL2\]](#) that are applicable to the SIF HTTPS and SIF HTTP Transport Layers, including associated encryption, signature and authentication algorithms, including the use of X.509 certificates.

- XML 1.0 [\[XML\]](#) and its references to Unicode and the UTF-8 Encoding.

- The role and use of namespaces in XML [\[XMLNS\]](#).

- Accessing XML elements/attributes using XPath [\[XPATH\]](#).

- XML Schema data types and structures [\[XMLSCHEMA\]](#).

- Relational database and message queue concepts.

It is furthermore assumed that implementers have at their disposal or can implement:

- Adequate XML tools (e.g. parsers; parsers that can validate using XML Schema, if desired; simple XPath evaluators) as they develop SIF-compliant software.

- Implementations of HTTP(S) that support the SIF HTTPS Transport Layer, and optionally, the SIF HTTP Transport Layer.

3.1.1 Notes on Related Technologies

Implementers are referred to

[Notes on Related Technologies](#), which highlights technologies leveraged within SIF or related to SIF, either in their entirety or as a subset. This partially normative appendix points out specifics casual readers of referenced documents on these technologies must not ignore when implementing SIF Zone Integration Servers and agents.

3.2 Architectural Components

A SIF Zone is a distributed networking system that consists of a Zone Integration Server (ZIS) and one or more integration agents. The size of a zone is flexible and could consist of a single building, school, a small group of schools, a district, a region, a state, a nation, etc. SIF is a scalable solution for data exchange. A SIF Implementation consists of one or more SIF Zones deployed and configured to meet customer data sharing and reporting needs.

A Zone Integration Server is the heart of the SIF architecture. The ZIS is a program that provides integration services to all the agents registered with it so that they can provide data, subscribe to events, publish events, request data, and respond to requests. It is responsible for all access control and routing within the Zone.

Each application requires an agent, which typically is provided by the application vendor, to communicate with other applications via the ZIS and their respective agents. For example, a school may use a student information application, a food service application, and a library automation application. Each of these applications must have an agent that acts as a go-between between the application and the Zone Integration Server.

In SIF, an agent never communicates with another agent directly. Instead, each agent communicates with the ZIS as a trusted intermediary that brokers the exchange of data with other agents. Having the ZIS manage routing responsibilities allows complex communications to occur between agents that have no direct information about each other and that may or may not be available for communication at any given point in time.

The following diagram illustrates a typical single-zone SIF implementation for a school.



Figure 3.2-1: Single-Zone School SIF Implementation

3.2.1 SIF Systems

A zone is often defined according to physical boundaries; for example, a zone can consist of all the applications that are connected over a private network and managed by one organization, such as a school. Security, scalability, and manageability requirements can also influence the decision of how zones are designed and configured.

Zones are a flexible and powerfully creative tool for meeting the data exchange and reporting needs of users; zones can be as varied as the customers in the education marketplace. While a single school zone may meet the needs of a single school, SIF implementations can scale to meet the needs of specific end users through the use of multiple zones, sometimes managed by different ZIS implementations. Two examples of many multiple-zone implementation design patterns are included here for illustration.

In the first, the data exchange needs of a district are met through the use of four zones, one for the district, and three for schools within the district: elementary, middle and high school. Here a student information system provides its complete set of district-wide data to a district zone, while providing school-based views of and access to that data in the individual school zones. Library systems in this implementation are school-based, while the food services system, like the SIS, is district-based.

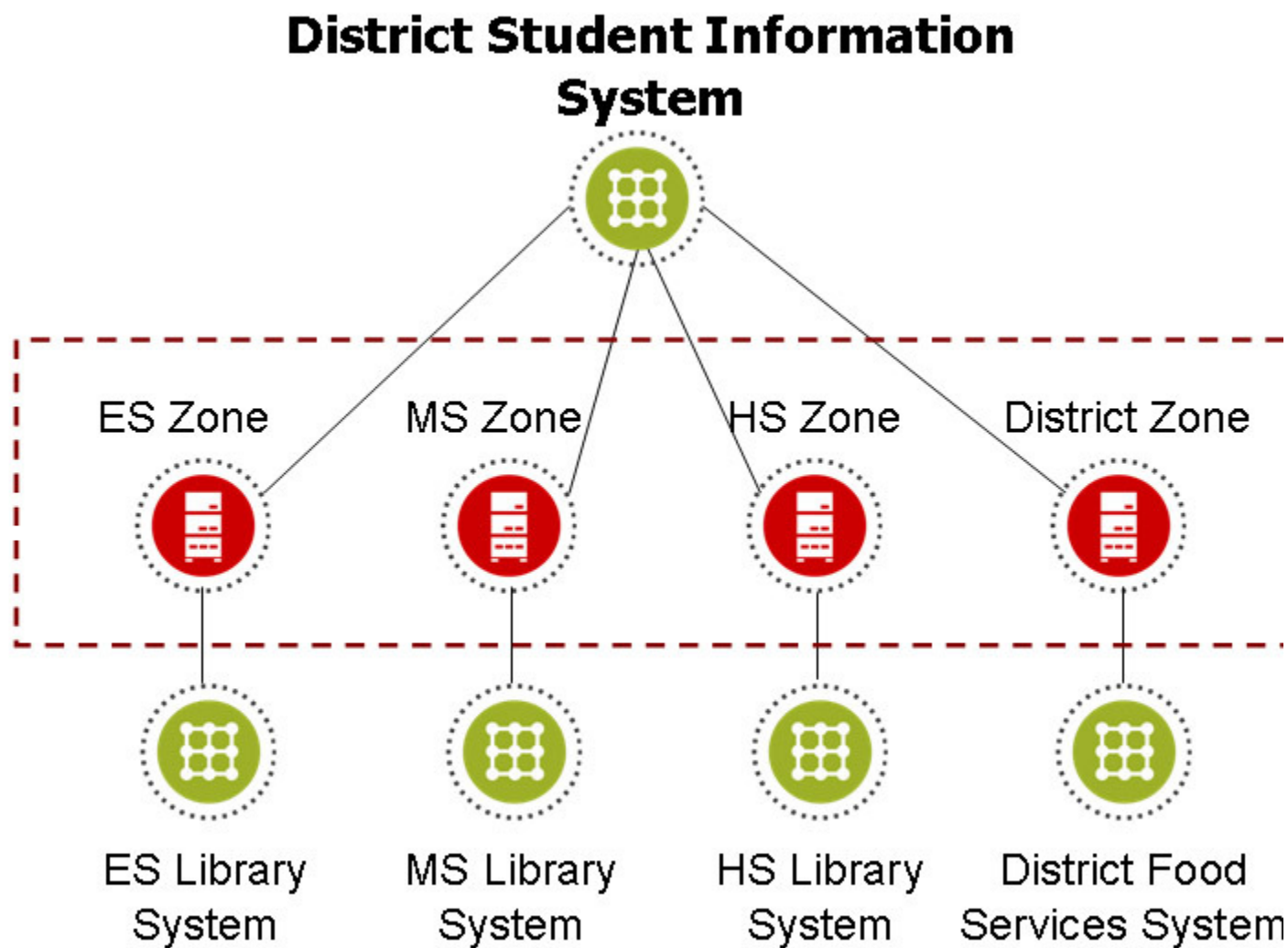


Figure 3.2.1-1: Multiple-Zone District SIF Implementation

The second example illustrates an agent communicating in both a district and a state zone. This agent could be associated with many different types of applications, including a SIS or data warehouse, reporting data up to the state, or an application that supports `StudentLocator`, managing state-level student identifiers, and so on.

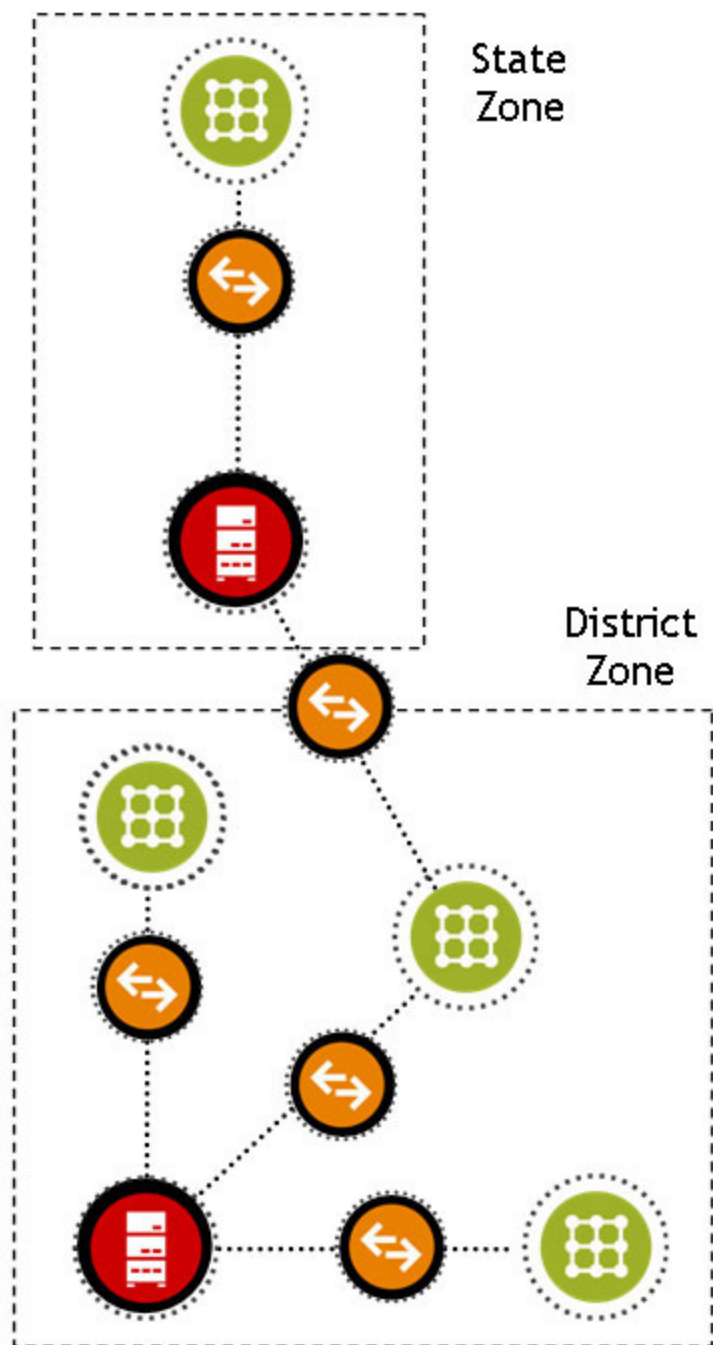


Figure 3.2.1-2: Multiple-Zone State SIF Implementation

Multiply the district portion of this diagram by dozens or hundreds of districts, each with its own local zone or configuration of zones, and the distributed scalability of SIF using zones is readily apparent.

3.3 Concepts

This section presents the ideas behind the implementation of SIF, including the application and data models on which it is based. It serves as a precursor to further descriptions in following sections.

3.3.1 Data Model

Data exchanged in SIF is defined using a series of *data objects*. These objects are expressed using

[XML] and are defined structurally by this document and associated schemas, with this document and supporting documentation defining the semantics behind the exchange of individual data objects such as: `StudentPersonal`, `StudentSchoolEnrollment`, and `StaffPersonal`.

3.3.2 Zone Architecture

The common feature of all SIF topographies is that a number of applications wish to share data. SIF implementations consist of one or more applications with their associated agents communicating via a Zone Integration Server (ZIS).

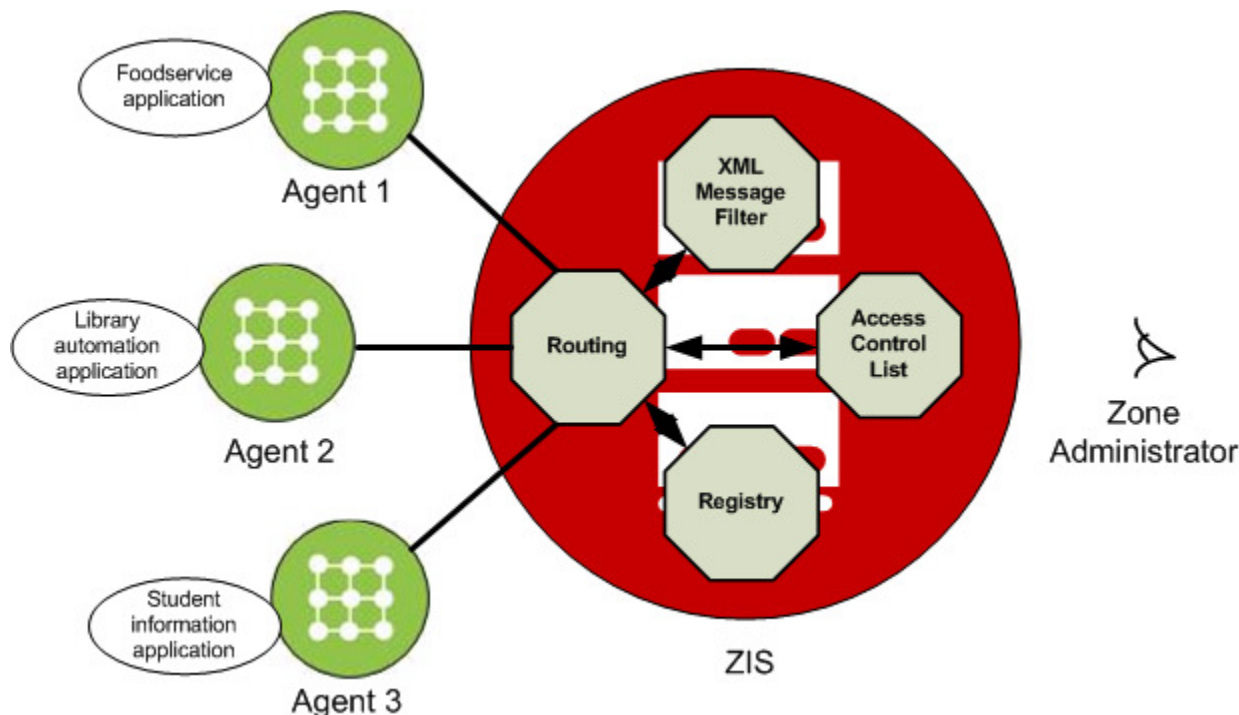


Figure 3.3.2-1: Zone Architecture Block Diagram

One typical use of SIF is to connect products from various vendors together within a single school. Typical applications include a student information application, a food service program, and a library automation application. Each of these applications are accessed through a vendor-provided interface program called an *Agent*.

Since the same school shares these applications, it makes sense to group them together into a logical entity. This entity is referred to as a zone and is managed by a Zone Integration Server (ZIS).

There are no predefined sizes for zones, so a zone can be as large or small as required in order to meet the needs of the end user.

An application relies on its agent to exchange data using predefined data objects. The ZIS enables agents to communicate by routing these data objects through the zone.

To avoid erroneous and potentially unauthorized access to information, the ZIS also provides access control so the Zone Administrator can control which applications have access to which data objects.

3.3.2.1 Contexts

The Zone is the primary means of partitioning data, applications, and policies. Zones are typically organized around geographic boundaries (e.g. school, district, region, state) or functional boundaries (e.g. horizontal integration, student locator services, data warehousing and reporting services). A SIF Context offers the ability to further partition the data within a Zone, that is, to offer different perspectives of the data based on customer needs and application abilities. For example, while a student information system typically serves as the source for student-related data in the default context of a zone, a data warehouse might better be suited to provide a historical or longitudinal perspective of that exact same student data in a different context, a context more suited to the reporting and data warehousing needs of an implementation. Contexts enable end users and system integrators to work with data in new ways while retaining the zone topologies commonly in use.

In addition to offering different perspectives on a zone's data, contexts allow two or more agents to register as a provider of the same object type within a zone. This may lead to future solutions built around contexts; for example, to better define how systems that publish similar objects cooperate in the same zone (e.g. student information systems and special education packages).

Contexts also make it easier to apply a different set of business rules to different audiences. Unlike zones, which can be named and assembled in a variety of ways at the discretion of system integrators, contexts are named in the Specification and are therefore part of the specification. The SIF Association sanctions contexts and provides documentation that defines each context's purpose and any associated message choreographies and business rules for it. Contexts defined by the SIF Association have context names that begin with `SIF_` and the default context for a zone is named `SIF_Default`. It is **RECOMMENDED** that all ZIS implementations support the SIF Association-defined contexts as they are introduced; support for user-defined contexts is strictly implementation dependent, and agents are discouraged from relying on ad hoc or user-defined contexts.

3.3.3 Infrastructure and Messaging

Agents share data in a Zone via two models, the Publish/Subscribe model and the Request/Response model. In the Publish/Subscribe model, agents publish data changes of interest to subscribers by sending a `SIF_Event` message to the ZIS.

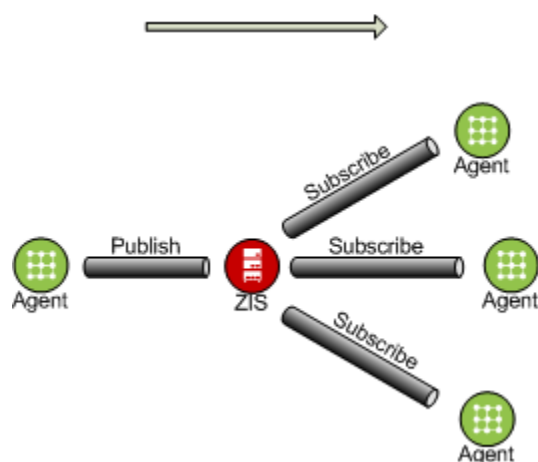


Figure 3.3.3-1: Publish/Subscribe Model

In the Request/Response model, agents request or query data from other agents in a Zone by sending a `SIF_Request` message to the ZIS, eventually being sent one or more `SIF_Response` messages from an agent in return.



Figure 3.3.3-2: Request/Response Model

This exchange of messages over a SIF-defined transport layer, SIF HTTPS or SIF HTTP, is the primary feature that defines the SIF Infrastructure.

Every message exchanged over this infrastructure is wrapped inside a `SIF_Message` and contains a `SIF_Header` element that specifies the source of the message and optional security, destination and context information. In addition to the messages exchanged between agents via the ZIS, the SIF Infrastructure defines a number of messages that are exchanged between agent and ZIS, and between ZIS and Push-mode Agent—these serve primarily to register various agent settings at the ZIS and to support the exchange of messages between agents.

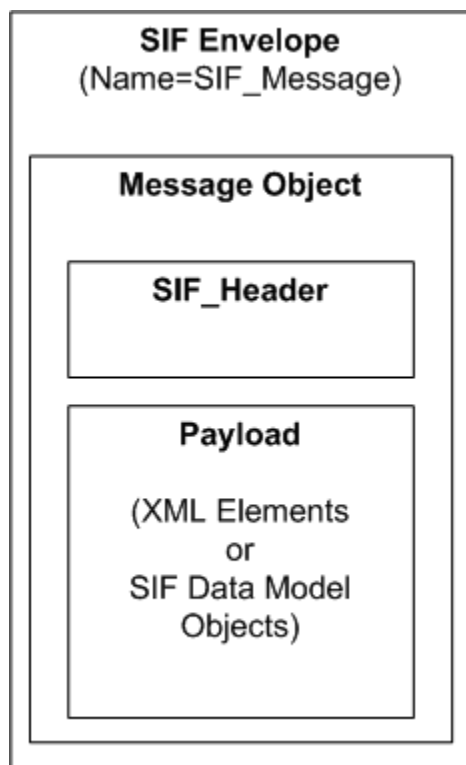


Figure 3.3.3-3: Message Structure

3.3.4 Data Provision: A Request/Response Model

3.3.4.1 Routing

When an application (the *Requester*) wants to gather data from a specific data object, a `SIF_Request` message is sent to ZIS. The application may direct this request to a given *Responder* by specifying an Agent Id in the `SIF_DestinationId` element of `SIF_Header`. In most cases, however, the `SIF_DestinationId` element is omitted in which case the ZIS routes the request to the default responder, or *Provider*, for the data object of interest. Agents register as Providers with the ZIS using either the `SIF_Provision` or `SIF_Provide` message.

There is a single Provider per object per context per zone. There may be multiple Responders for a given object in a zone context.

3.3.4.2 Access Control

In order to maintain control over what data is exchanged over the zone and who exchanges it, the ZIS must provide an access control system that limits who can provide, request, and respond to requests for which data objects. The access control system must maintain policies for each registered application.

If the requester knows or wants to control who the responder will be, it must place the responder's agent identifier in the `SIF_DestinationId` element of the header of the `SIF_Request` message. The ZIS will examine the `SIF_Request` message's header. If a `SIF_DestinationId` element is present, the ZIS must route the `SIF_Request` to the specified agent/application subject to the limitations imposed by the access control security policies for the zone. For instance, even though an application specifies that it wishes a specified application to respond, the zone security policy may prohibit the specified application from generating `SIF_Response` messages.

An application that wants to provide access to the data it contains via SIF may function as a responder. Such applications will support one or more SIF data objects. The application listens for `SIF_Request` messages for the objects that it supports. When it receives a `SIF_Request` for a supported object, the application will generate one or more `SIF_Response` messages containing the application's data, which will be routed by the ZIS to the requester. The responder must place the requester's agent identifier in the `SIF_DestinationId` element of the header for each `SIF_Response` message generated.

3.3.4.3 Error Reporting

When an application receives a `SIF_Request` for a data object that it does not support, it must return a `SIF_Response` message with the `SIF_Error` element populated to indicate the nature of the error (invalid object), a `SIF_PacketNumber` of 1 and the `SIF_MorePackets` element set to indicate that no further packets will be sent in response to the `SIF_Request`.

3.3.5 Event Reporting: A Publish/Subscribe Model

Applications propagate data updates by publishing `SIF_Event` messages for the SIF data objects that are being added, changed, or deleted. In order for an application to receive these `SIF_Events`, subscriptions for the SIF data objects of interest must be entered at the ZIS. This subscription process is performed when an application sends a `SIF_Provision` message or one or more `SIF_Subscribe` messages to the ZIS. Once the subscriptions are entered, any `SIF_Events` for those objects received by the ZIS will be routed to the list of subscribers for those objects.

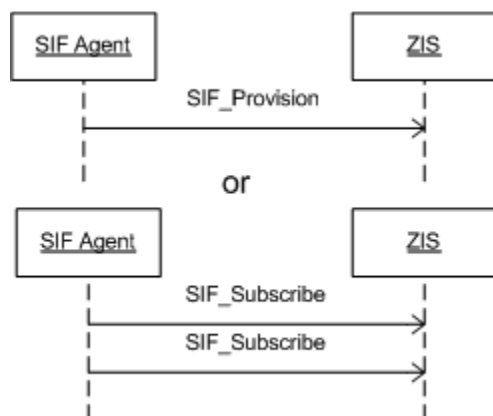


Figure 3.3.5-1: Subscribe to Events

Once an application successfully sends a `SIF_Event` to the ZIS, the ZIS is responsible for delivering that `SIF_Event` to the subscribing parties without any further communication to the `SIF_Event` originator. The `SIF_Event` originator does not know how many applications, if any, receive the `SIF_Event`. No notifications are provided to the originator to indicate whether a `SIF_Event` was delivered to a subscriber or not.

The ZIS must maintain an access control system that limits who can publish and subscribe to events for which data objects.

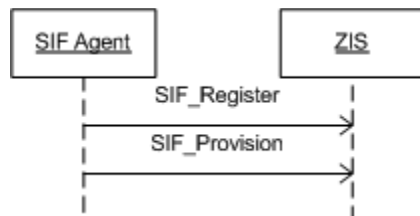


Figure 3.3.5-2: Register with ZIS

Before an application can utilize the services of the ZIS, the application must register itself by sending a `SIF_Register` message to the ZIS. Once registered, an application does not have to perform any additional registration with the ZIS in order to be a publisher of `SIF_Event` data. Any application that has registered itself with the ZIS may publish `SIF_Events` subject to the limitations imposed by the access control security policies for the zone. It is recommended that event publishers register their ability to publish events by using the `SIF_Provision` message.

Multiple applications may publish `SIF_Event` messages for a given data object.

The application that is registered as the Provider for a given data object must be able to subscribe to `SIF_Events` for that object but the application is not required to subscribe to `SIF_Events` in a given SIF implementation.

An application that has subscribed to a `SIF_Event` must attempt to process the `SIF_Event` according to the business rules of the application. If the `SIF_Event` contains insufficient information or information that is inconsistent with the application's business rules, the application may ignore the message.

If an application publishes a `SIF_Event` as a result of changing the data within the application and the ZIS rejects the `SIF_Event` message, it is recommended that the application rolls back or cancels the changes that were made, but the application does not have to roll back the changes. For example, an application may attempt to add a new student and publish a `SIF_Event` to reflect the

addition. If the application does not have permission to publish `SIF_Event` messages for that type of object, the `SIF_Event` is rejected. The application does not have to remove the newly added student from its local database.

3.3.6 Communication: An Asynchronous Model

In order to ensure scalability and reliability, SIF requires that its request/response and publish/subscribe models be asynchronous in nature. Once a ZIS synchronously acknowledges receipt of a `SIF_Event`, `SIF_Request` or `SIF_Response` with the return of a successful `SIF_Ack`, an agent cannot be assured that these messages will immediately be delivered to subscribers, providers/responders or requesters, respectively, or that it will receive an immediate `SIF_Response` to any submitted `SIF_Request`.

The asynchronous communication model can be likened to communicating with someone via e-mail or through the postal office: an individual sends the message, but does not know when it will be received, much less when the receiver will respond.

3.3.6.1 Guaranteed Delivery

In contrast to the asynchronous communication model, most agent-to-ZIS and ZIS-to-agent communication—over currently defined transport layers—is synchronous in nature. Any time an agent sends a `SIF_Message` to a ZIS, the agent waits for a `SIF_Ack` to be returned from the ZIS to acknowledge receipt of the message. Once acknowledged, the ZIS guarantees future delivery of `SIF_Event`, `SIF_Request` and `SIF_Response` messages, barring certain error conditions. For messages not directly related to the request/response and publish/subscribe models, the acknowledgement from the ZIS also indicates successful completion of operations related to registration, subscription, provision and system control operations. To complete the guaranteed delivery, when a ZIS contacts an agent in Push mode, the ZIS waits for a `SIF_Ack` to be returned from the agent to acknowledge successful delivery of the message currently pending for the agent.

3.3.7 Security Model

The security model of SIF centers around three areas: encryption, authentication and access control. SIF provides application agents the ability to specify the encryption and authentication requirements for all other agents that eventually come into contact with their sensitive data. Various communication protocols over which SIF data may be transferred, including SIF HTTPS, provide built-in support for easing the implementation details of guaranteeing encryption and authentication requirements. In addition, access control at the ZIS allows a zone administrator complete control over which agents are allowed to communicate which data to other agents.

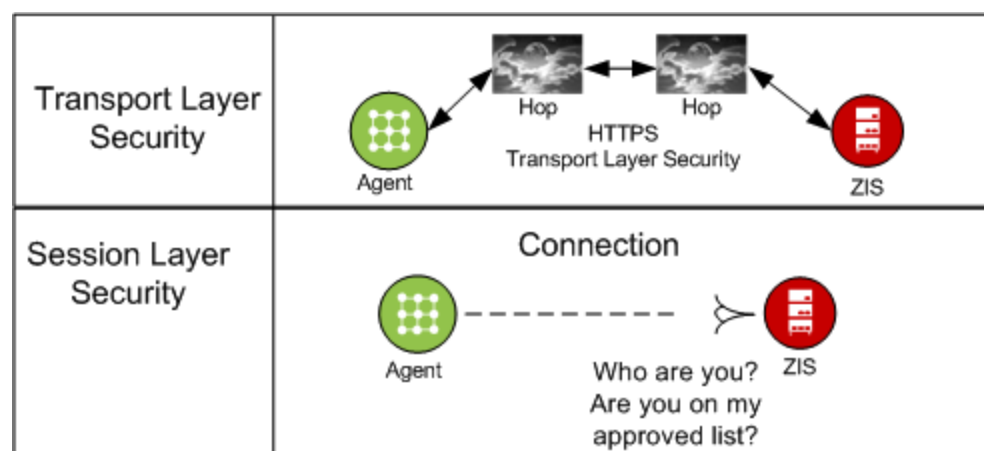


Figure 3.3.7-1: Security Model

As SIF HTTPS is the default communication protocol that all agents and ZIS implementations must support, many of the encryption and authentication levels specified in this document are tailored to the encryption and authentication algorithms currently defined within SIF HTTPS. When a ZIS implementation supports other communication protocols, the ZIS must guarantee that these levels are accurately reflected and adhered to when communicating with agents that support these same protocols.

3.3.7.1 Encryption

Encryption provides the mechanism to ensure that only the sender and receiver of a message can view the message contents. In a totally secure model, all communications between agent and ZIS will be encrypted. The SIF HTTPS protocol, which must be supported by all agents and ZIS implementations, is a secure transport and provides encryption of the data being exchanged.

If additional communication protocols, or transports, are used, it is important to know if these transports are secure to avoid exposing sensitive data. SIF provides a method for an agent to specify to the ZIS how secure the channel between the ZIS and other

agents must be when ultimately delivering the originating agent's sensitive data. ZIS implementations must guarantee the requested security levels when communicating with recipient agents, regardless of which transport is in use. If a ZIS is unable to ensure these security levels when communicating with a recipient agent, the ZIS must not transport the message across the insufficiently secure channel. It is recommended that the ZIS log the inability to deliver the message to the recipient agent due to security requirements.

The responsibility for guaranteeing the security of data that an originating agent transfers to the ZIS lies ultimately with the originating agent, or zone administrators. For example, if the originating agent requires a very secure channel for a given message, it should not intentionally or inadvertently communicate that message to the ZIS over an insecure or insufficiently secure channel, should the ZIS support such channels. At that point, the data has already been communicated insecurely. Zone administrators can prevent such occurrences by configuring the ZIS and agents within the zone such that a minimum security level is maintained, below which communication is impossible.

In many cases, the establishment of a secure channel and encryption can be delegated to the transport layer.

3.3.7.2 Authentication and Validation

The role of authentication is to provide a means to ensure that the author of a message is the actual author. Authentication guards against a situation where a foreign agent claims to be a legitimate zone participant and fakes a message to gain access or alter the SIF data, i.e., spoofing.

Another important role of authentication is to provide the ability to detect that each message that passes through the Zone arrives at its destination unaltered and unread by other intermediaries, i.e., man-in-the-middle attacks.

Authentication support is optional but highly recommended.

3.3.7.3 Access Control

SIF Zone access **MUST** be able to be controlled centrally at the ZIS, allowing for local administration of Zone security policies.

A SIF administrator **MUST** be able to specify which applications **MAY** participate in the SIF Zone, which data objects each application **MAY** provide or request, and what events each application **MAY** produce and receive. Refinements in the granularity of control are permissible. In addition, a SIF administrator **MAY** be able to specify XML filter rules that remove messages or specific XML elements or attributes before being delivered to the application.

The access control requirements are discussed more fully under ZIS Requirements.

3.3.8 Zone Services

Until SIF Implementation Specification version 2.4, the SIF Zone only allowed applications to interoperate by exchanging messages conformant with the SIF data model, in accordance with one of the two data exchange models (Request / Response or Publish / Subscribe) defined above. In a sense this was equivalent to constraining application interaction to what would be possible if they shared a reliable, secure common data store and an associated set of record schema, and were automatically notified via a database trigger whenever a partner updated the data.

While this represented a powerful way to unify remote applications, the following capabilities were not provided:

Customized Operations: Create/Read/Update/Delete (CRUD) data were the only requests possible.

Transactions: Reliably know what, if anything, happened as a result of sending an update event.

Database Views: Package up and send elements from related data objects together in one message.

All these capabilities are offered by the traditional service paradigm, where the internals (such as the object hierarchy and process sequencing) are hidden in the implementation, while the client sees only the service interface.

In a service paradigm, individual services are choreographed by other services to provide new capabilities. Zone Services are the way in which these capabilities may be realized within the SIF Zone. The key design constraint in its development was that the underlying SIF infrastructure would be extended (via additional message types) and not replaced. Zone Service clients are able to communicate with Zone Services over the same wire that SIF Object clients communicate with Object Providers, and they operate in much the same way. All of them are fully supported Zone citizens, and like previous SIF components, will generally consist of an agent and an application.

Clients must be able to invoke methods on Zone Services.

All Service invocations are asynchronous.

Clients and Zone Services never communicate directly with each other.

Message exchange patterns are identical to the ones that exist for object providers.

Provisioning/System Messaging

Request/Response

Publish/Subscribe

As a result, the extended infrastructure functionality supports both the invocation of specific Zone Service Methods (with defined arguments), and the notification of Zone Service events to service subscribers. By providing this service capability within the SIF Zone, the normal message functionality of the Zone applies to the new message types as well:

Data security (via data encryption, authentication and specific administrator authorization)

Loose coupling between sending and receiving application (a service need never know its subscribers)

Guaranteed message delivery or failure notification

Guaranteed correct packet ordering on reception

Automatic service discovery

Content based routing

All data exchanges map back to the underlying SIF Data Model

Data Object	Zone Service
Data elements are adjectives describing the object.	Operations are verbs that describe the actions a service can perform.
Models an entity (e.g., Student).	Models a process (e.g., Locate Student).
Stateless data	Stateful behavior
Single fixed CRUD interface	Customized interface
Single owner (provider) per object per context within a Zone.	Multiple Zone Services may supply or change a given object (usually by implementing calls to the Object Provider).
Allows applications to synchronize their data sets.	Allows applications to interact at a deeper level.
Multiple applications besides the Object Provider can publish change events for a given object.	Only the default Service Provider can publish notifications for the service.
Selective Message Blocking (SMB) can block object events.	Selective Message Blocking (SMB) will not block Zone Service notifications.

Table 3.3.8-1: Differences between a Data Object and a Zone Service

3.3.9 Naming Conventions for Agents and Zone Integration Servers

SIF requires that each agent and ZIS be identified with a distinct case-sensitive identifier that is unique within a zone. This identifier is carried inside the `SIF_SourceId` element of the `SIF_Header` included in each SIF message and is used, among other things, at the ZIS to reference access control permissions of each agent within the zone. It is recommended that agent and ZIS implementations have user-configurable identifiers in order for zone administrators to maintain unique identifiers within the zone.

The identifier should be descriptive of the role of the application in the zone. For example, the library automation agent for Ramsey Elementary might carry the identifier `RamseyLib` instead of the less descriptive `CC41Agent`. The Zone Integration Server for Ramsey might be known as `RamseyZIS`.

3.3.10 Object Identifiers

Data objects and the data local to an application that map to these objects often must be retrieved by a unique identifier. Likewise there often exist relationships between data objects that require a unique key or identifier for efficient look-up of related data. SIF provides these keys or unique identifiers through object identifiers, also known as RefIds or GUIDs thanks to SIF naming conventions and the type of identifiers used in SIF, respectively. The `StudentPersonal` object, for instance, carries detailed information about a student, and most agents that manage or require student information reference the data stored in this object and

often map the `RefId` of `StudentPersonal` to locally stored data, or request `StudentPersonal` objects from the zone by `RefId`. Objects often carry an attribute that identifies a particular object instance; this attribute is named `RefId`. It is imperative that `RefId`s not clash with any other `RefId`. This is especially relevant when an agent manages a database comprised of a mix of objects; for example, a library database containing patrons, which are a mix of both students and staff. To virtually eliminate the possibility of duplicate object identifiers and to provide a consistent, decentralized way of generating these identifiers, SIF requires the use of a globally unique identifier (GUID) that **MUST** be generated per published algorithms

[RFC 4122] whenever a `RefId` is used. GUIDs in SIF have their own format; they **MUST** be 32 characters long and contain only valid upper-case hexadecimal characters (0-9, A-F) with no spaces or punctuation.

Object identifiers do not have to appear on any customer screens and they do not replace any identifiers currently in use by applications. Applications and application users can still reference data as they always have. The GUID provides an additional key, which becomes the SIF primary key that agents use to reference an object within SIF.

As stated, object identifiers are also used to represent relationships between objects. Where referenced, `RefId` is typically prefixed with the object name, e.g. `StudentPersonalRefId` in `StudentPicture` refers to the `RefId` of the `StudentPersonal` object corresponding to the student photographed e.g. `LearnerPersonalRefId` in `LearnerSchoolEnrolment` refers to the `RefId` of the `LearnerPersonal` object corresponding to the enrolled learner. Other more complex conventions surrounding object identifiers and `RefId`s can be found in [Data Model](#).

3.3.10.1 Object Identifier Persistence

When used as identifiers for objects that persist over time—take for instance the `StudentPersonal` object that represents a student in a Zone—it is SIF's intent that object identifiers not change over time. The `RefId` attribute for John Doe in first grade should have the same value when John Doe is in second grade, in middle school or in high school. This persistence of object identifiers enables longitudinal tracking of data within SIF, especially where there exist no locally unique identifiers associated with objects. Implementations **SHOULD** avoid reassignment of object identifiers within a zone and as the primary home for individual objects may move from one zone to zone over time (e.g. a student moving from a middle-school to a high-school zone).

3.4 Agent/Application Requirements

Each application that wants to be a SIF application, or SIF-enabled application, must have an agent written for it. An agent is an extension to the application that communicates with the ZIS. An agent can be an integral part of an application itself, or may be a separate, specialized client of or interface to an application.

All applications that are part of a SIF zone must be able to gracefully handle all SIF messages including those messages and data objects that the application does not support. It is **RECOMMENDED** that the application return an error `SIF_Ack` message to the ZIS for those messages that the agent does not support (error category Generic Message Handling, error code "Message not supported"). An agent **MAY** return an "Immediate" `SIF_Ack` to the ZIS and ignore unsupported messages.

High-level functional requirements for all SIF-enabled applications include the following. More detail on particular requirements may be found in

[Messaging and Infrastructure](#).

3.4.1 Communicate with the ZIS

3.4.1.1 HTTPS

Support for

SIF HTTPS is **REQUIRED** of all agents. An agent **MUST** be able to communicate with the ZIS using SIF HTTPS, but it may attempt to communicate with the ZIS using any communication protocol defined in this or other specifications. **SIF HTTP** is the other communication protocol defined in this specification at this time. Support for any communication protocol other than SIF HTTPS is implementation-dependent. If connection attempts in protocols other than SIF HTTPS fail, a connection over SIF HTTPS should be made in order for communication to proceed. Given the sensitive nature of much of the data within the zone, it is **RECOMMENDED** that all communication occur over SIF HTTPS or similarly secure communication protocols.

3.4.1.2 Agent Registration

Given a communication channel between agent and ZIS, an agent is **REQUIRED** to register with the ZIS if it is not already registered or if it wishes to change or retransmit its registration settings. The `SIF_Register` message provides the ZIS information regarding agent capabilities and requirements, and allows the ZIS to contact the agent in the future if the agent is capable of accepting ZIS-initiated communications (a Push-Mode Agent).

An agent **MAY** also indicate its support for various data objects and associated messages using one or more of the `SIF_Provision`, `SIF_Provide` and `SIF_Subscribe` messages.

3.4.2 Transmit Application Changes to the ZIS

When an application makes changes to its data that correspond to a SIF object it supports, the application **MUST** be able to publish `SIF_Events` reflecting those data changes. If the application/agent makes changes to its data in processing a `SIF_Event` it has received, it **MUST NOT** publish an event that duplicates the changes as described in the processed `SIF_Event`. However, should the application/agent make additional changes beyond those in the `SIF_Event` being processed, the application **SHOULD** generate a new event describing the additional changes.

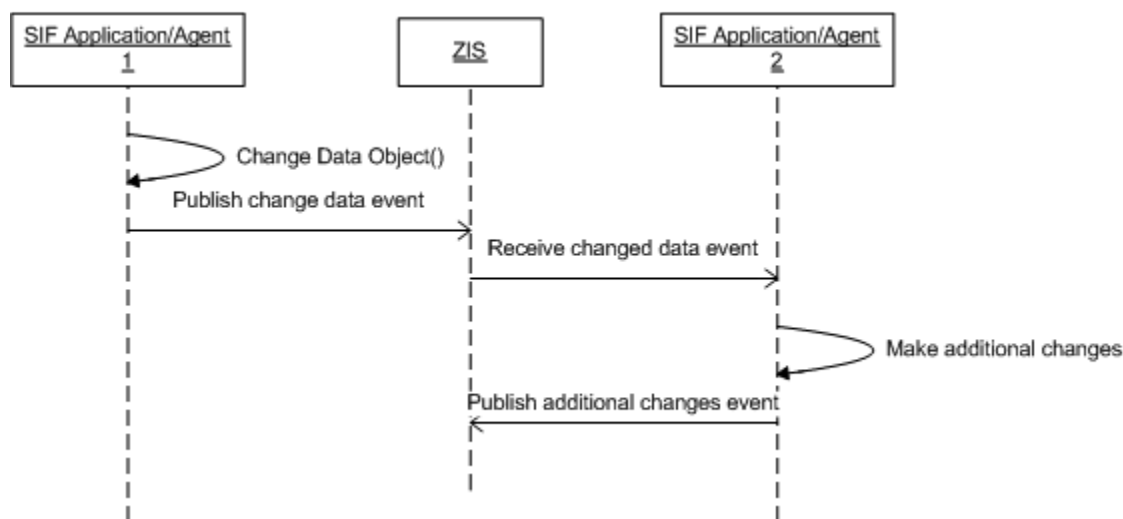


Figure 3.4.2-1: SIF Event

If an application does not support an optional field of an object or an element is not part of the change being reported by a `SIF_Event`, the application **MUST NOT** send an empty element, (e.g., `<BirthDate/>` or `<BirthDate></BirthDate>`); it **MUST** omit the element from the XML stream instead.

When publishing `Add` events, agents **MUST** include all elements listed as mandatory in

[Data Model](#) for the object.

3.4.3 Respond to Requests

All agents **MUST** be prepared to handle `SIF_Request` messages for all objects gracefully. In the case where an agent receives a `SIF_Request` for an object that it does not support, in addition to acknowledging the receipt of the message to the ZIS it **MUST** send a `SIF_Response` message to the Requester with the `SIF_Error` element populated to indicate the nature of the error (invalid object), a `SIF_PacketNumber` of 1 and the `SIF_MorePackets` element set to indicate that no further packets will be sent in response to the `SIF_Request`.

If an application agent is a responder for any object, the agent must be prepared to process `SIF_Request` messages for that object. This involves the ability of the agent to traverse the application database and construct an XML response stream based upon the parameters of the query request. All responders **MUST** support `SIF_Query` and query conditions that reference root attributes of the object as well as any mandatory elements within the object, along with their mandatory attributes. Responders **SHOULD** support query conditions that reference optional elements and their attributes, when the application supports such queries. Responders **MAY** support `SIF_ExtendedQuery` and **MUST** register their support for `SIF_ExtendedQuery` using `SIF_Provision` and/or `SIF_Provide`.

When an agent is creating `SIF_Response` packets, it **MUST** attempt to ensure that each packet is no larger than the `SIF_MaxBufferSize` specified by the `SIF_Request`. If for any packet a single packet does fit within the supplied `SIF_MaxBufferSize`,

the agent **MUST**, in addition to acknowledging receipt of the message to the ZIS, send a `SIF_Response` message to the Requester with the `SIF_Error` element populated to indicate the nature of the error, and the `SIF_MorePackets` element set to indicate that no further packets will be sent in response to the `SIF_Request`.

The `SIF_Request` message also contains `SIF_Version` elements that specify which SIF versions the responding agent should use when preparing the response packets. If a responding agent can support a single requested SIF version, it returns a response packet using that version. If more than one version is specified and the responding agent supports more than one of those versions it **SHOULD** respond with the highest version it supports. If the agent cannot support any requested SIF version, in addition to acknowledging receipt of the message to the ZIS, the agent **MUST** send a `SIF_Response` message to the Requester with the `SIF_Error` element populated to indicate the nature of the error, a `SIF_PacketNumber` of 1 and the `SIF_MorePackets` element set to indicate that no further packets will be sent in response to the `SIF_Request`.

If any other error occurs while creating `SIF_Response` packets for a given request, in addition to acknowledging receipt of the message to the ZIS, the agent **MUST** send a `SIF_Response` message to the Requester with the `SIF_Error` element populated to indicate the nature of the error, with `SIF_MorePackets` set to indicate that no further packets will be sent in response to the `SIF_Request`.

Agents supporting `SIF_Requests` **MUST** be able to return all of the object fields that the responding application supports or a subset of the fields as specified by the query request. For example, an agent may request that only a student's graduation year be returned and not the entire `StudentPersonal` object. If the responder does not support a requested element, it **MUST NOT** exclude the object from the response stream. Any requested element that is unsupported is omitted from the response stream; when processing `SIF_Query` requests, parent elements of requested elements, including the object itself, are included in the response stream.

If an application does not support an optional element of an object, it **MUST NOT** return an empty element. The element **MUST** instead be omitted from the XML stream.

3.4.4 Vendor Application Support for SIF

Depending upon the type of architecture, the core application may need to be altered to ensure that the agent is able to forward changes to objects of interest to SIF. For example, an application that edits student data may need to be modified to capture the adds, changes, and deletes made to students and store them into a temporary repository until the agent can forward them to the ZIS. Other architectures provide the ability to trap these changes at a server level eliminating the need for any changes to the application itself.

To meet the SIF requirement of data robustness, it is highly **RECOMMENDED** that all changes to objects of interest to SIF be persisted using a database table, local message queue, or other highly reliable storage system. This specification allows for the ZIS and any or all agents to be offline at any given time. Without storing agent changes locally, these changes can be lost when the ZIS is temporarily unavailable; local storage allows these changes to be transmitted to the ZIS when it becomes available.

When an object is shared for the first time in SIF, it is the responsibility of the application making the object available to assign its object identifiers/primary keys, typically a `RefId` in the form of a GUID, before releasing that object to the zone in an `Add` event or in a `SIF_Response`. Some application databases are extended to include SIF object identifiers; others maintain mappings from SIF object identifiers to locally-defined keys.

If an application changes data that maps to a SIF object, it is **RECOMMENDED** that only the changed fields be sent to the zone. This will result in smaller message sizes and improved performance.

To avoid unintentional overwriting of data, unsupported fields or fields that have not been changed **MUST NOT** be sent to the zone using empty XML elements, (e.g. `<Name Type="04 "/>` or `<Name Type="04 "></Name>`); the fields **MUST** be omitted from the XML stream instead.

3.4.5 Support Authentication and Digital Signatures

Supporting authentication is not a requirement but it is highly **RECOMMENDED** to ensure that your agent will be able to communicate with any ZIS. SIF does not mandate the use of authentication, but it is feasible that many SIF implementations will require this functionality. This is especially true for installations that may use the Internet to transport data.

Typically the authentication and verification mechanisms that are built into the network operating system or transport protocol can be leveraged. If these services are available, authentication and verification take place completely within the underlying security package.

The SIF HTTPS protocol supports authentication between an agent and a ZIS. If authentication is enabled and properly configured, a message receiver (agent or ZIS) can trust the SIF HTTPS implementation to verify that the message in its entirety comes from the claimed sender.

3.4.6 Agent Local Queue

An agent can be developed with a local queuing mechanism so that it can automatically cache incoming messages in a local queue and can acknowledge receipt of each message to the ZIS with "Immediate" `SIF_Ack` messages (which causes the ZIS to remove received messages from the agent's queue). Agents with an Agent Local Queue do not need to send any "Intermediate" `SIF_Acks` to the ZIS. Use of an Agent Local Queue can be used to locally support selective processing of messages, similar to the functionality provided by Selective Message Blocking; its use also allows more flexibility and robustness during application/system failure when successfully acknowledging events, requests and/or responses before performing the corresponding `SIF_Event`, `SIF_Request` and/or `SIF_Response` handling.

Thea Agent Local Queue is not a required feature of any agent. Agent developers can choose not to develop the Agent Local Queue mechanism.

3.4.7 Wildcard Version Support

It is possible for a SIF Zone to contain agents written to different versions of the SIF Implementation Specification. If a ZIS supports multiple versions in a Zone and there is at least one version in common with all registered agents, they may then communicate with each other. It is also possible for a SIF Zone to contain agents that have no versions in common with other agents. These agents consequently have no ability to exchange `SIF_Event`, `SIF_Request` or `SIF_Response` messages, unless the ZIS provides message conversion as described in

"Multiple Version" Zones.

As message conversion is an implementation-dependent feature of a ZIS, it is **RECOMMENDED** that agents register in Zones and request data using `SIF_Version` wildcards (see [SIF_Register](#) for format) that allow for the exchange of data between agents supporting any subset of releases within a major release cycle of this specification (e.g. `2.*` or `*` to accept any `SIF_Message` in the 2.x lifecycle). (Note that `*` allows messages from any major version to be delivered, which can be structurally quite different across major versions and pose development challenges, and is not particularly recommended for indicating the ability to receive messages from all versions within a major version release cycle.) This maximizes the ability of agents to exchange messages and data in these zones and, for customers, maximizes the utility of zones supporting different versions of this specification.

Wildcard version support is particularly important for SIF-enabled applications that are not updated with each release of this specification. Furthermore, given that releases of the SIF Implementation Specification are on a more rapid release cycle beginning with version 2.1, typically smaller in scope than SIF Implementation Specification releases historically, it is anticipated that it will become more common for SIF-enabled applications in Zones to support different specification versions, and for more applications not to be updated with each release of this specification. Wildcard version support also allows applications to be SIF-enabled at any time in a SIF Implementation Specification major release cycle without risking the need to upgrade with the introduction of a new minor release of the specification, particularly when the new functionality offered by the specification does not apply to or impact the application.

Ignoring revision releases, the changes typical of releases within a given major version are limited to new data objects and optional additions to existing data objects (and optional infrastructure additions). This nature of a lower release being a subset of each higher release within a major release lifecycle—and of a higher release being a superset of each lower version—allows SIF-enabled applications access to the same elements they rely on at the time of their implementation from SIF messages defined by a number of SIF Implementation Specification versions. For associated implementation notes, see [Wildcard Version Support Implementation Notes](#).

While wildcard version support in this specification is only **RECOMMENDED**, SIF-enabled application developers should be aware that this support may be mandatory in some SIF Certification Program product standards [\[SIFCertification\]](#) associated with a major release cycle, if application vendors wish to establish their applications as SIF Certified™.

3.5 Zone Integration Server Requirements

The Zone Integration Server is the central integration point for all the agents in a zone. Depending on the message type, a ZIS either saves information in the messages or forwards the messages to other appropriate agents.

The ZIS implementer is free to internally manage registration and access permissions information in any form that the implementer supports. In order to provide an example of how an administration system may be structured, the sections below describe a database consisting of an Access Control List and a Zone Status.

3.5.1 Access Control List

Access control is needed to ensure that the information available in SIF only originates from, and is only accessible to, authorized agents. A ZIS *MAY* maintain access control on whether a zone administrator has granted an agent permissions to register.

A ZIS *MAY* exhibit behavior with regard to the ACL that could be perceived by an agent as if virtual tables exists defining the following information:

Field	Comments
Agent Id	The unique Id for an agent (provided as the Source Id in a <code>SIF_Register</code> message)
Register	May this agent register in the zone?

Table 3.5.1-1: Register

An example of this virtual table, which defines which agents are allowed to register in the zone, might be as follows:

Agent Id	Register
RamseySIS	true

Table 3.5.1-2: Virtual Table Example (Register)

In addition, a ZIS *MUST* exhibit behavior with regard to the ACL that could be perceived by an agent as maintaining per-context/per-object permissions for each message associated with SIF's Publish/Subscribe and Request/Response models. When an agent tries to inquire about a student's personal information, for example, the ZIS needs to check if the agent has the proper permission to request such information.

Field	Comments
Agent Id	The unique Id for an agent (provided as the Source Id in a <code>SIF_Register</code> message)
Context Name	The name of the SIF Context to which the permissions apply
Object Name	The object being manipulated (e.g., <code>StudentPersonal</code> , etc.)
Provide	May this agent register as the provider for this object in this context?
Subscribe	May this agent register as a subscriber for this object in this context?
Publish "Add" Event	May this agent publish "Add" events for this object in this context?
Publish "Update" Event	May the agent publish "Update" events for this object in this context?
Publish "Delete" Event	May the agent publish "Delete" events for this object in this context?
Request	May this agent request this object in this context?
Respond	May this agent respond to a request for this object in this context?

Table 3.5.1-3: Access Control

It is important to understand that this is a virtual table, defining the appearance of the functionality to the agents. The actual implementation of this functionality is at the discretion of the implementers of a ZIS. An example follows:

Agent Id	Context Name	Object Name	Provide	Subscribe	Publish "Add" Event	Publish "Update" Event	Publish "Delete" Event	Request	Respond
RamseySIS	SIF_Default	StudentPersonal	true	true	true	true	true	false	true
RamseySIS	SIF_Default	LibraryPatronStatus	false	false	false	false	false	true	false
...

Table 3.5.1-4: Virtual Table Example (Access Control)

In addition to access control permission violations, attempts to register any of this functionality with the ZIS may fail due to other reasons; e.g. unsupported transport mechanisms, there already being a provider for an object, etc. As a result, an agent **SHOULD** be able to gracefully handle corresponding error conditions or report those errors to a zone administrator.

3.5.2 Zone Status

The ZIS **MUST** maintain the status of the zone for implementation purposes, as well as for communicating this status to other agents, as defined in `SIF_ZoneStatus`, when requested. This status includes but is not limited to:

- product information about the ZIS;
- supported transport protocols, authentication methods and SIF versions;
- supported contexts (see below for more information);
- the currently registered agents, along with applicable registration settings and the current state of each agent; and
- lists of currently registered providers, subscribers, publishers, responders, and requesters.

Providing examples of virtual tables that illustrate storage of all the information associated with `SIF_ZoneStatus` is beyond the scope of the specification; implementers should refer to

`SIF_ZoneStatus` for requirements.

3.5.3 SIF XML Filter

A ZIS **MAY** maintain a list of XML filter rules that are applied to messages being delivered to individual agents. If enabled, the filters **MAY** instruct the ZIS to remove the specified elements or attributes from any SIF message containing such elements before placing the message in the recipient agent's queue. The filter **MAY** also remove the message which would not be delivered to the recipient agent's queue. How these filters are configured using the ZIS user interface is left up to the ZIS implementation. However, if this feature is supported by a ZIS, at a minimum, the SIF administrator **MUST** be able to set XML element and attribute filters on any optional element or attribute within the SIF data model including the document element `SIF_Message` and object elements in `SIF_Response`. (N.B.: Although this ZIS feature is currently optional, some locales (e.g., the UK) may require the feature to be present; and that all elements and attributes be subject to filtering.)

Consistent with the broader scope of privacy and security practice, no notice of the ZIS's message removal or modification is transmitted inside or outside the Zone, although a local logging of such activity by the ZIS is appropriate and **RECOMMENDED**. Both the original message and the changed message **MUST** be capable of being logged or stored by the ZIS in such a manner that a ZIS administrator with an appropriate security clearance can see both copies of the message.

In order to more easily support a future import and export format that will be defined, it is **RECOMMENDED** that the implementation within the ZIS allow for the XML filters to be specified using an XPath

[**XPATH**] syntax. An example of a set of XML filters follows. It is important to understand that this is a virtual table, defining some of the functional elements which may be specified by an import/export file in the future. The actual implementation of this functionality within the ZIS is at the discretion of the implementer.

In this example, the `MedicalAlertMessages` and `IDEA` elements are removed from `StudentPersonal` before being sent to the `AcmeLibrary` agent. Also, any `SIF_Event` messages from `AcmeLibrary` have the `StudentPersonal/LocalId` element removed before being delivered to `AcmeSIS`.

Agent Id	SIF XML Filter
AcmeLibrary	<code>//StudentPersonal/MedicalAlertMessages</code>
AcmeLibrary	<code>//StudentPersonal/IDEA</code>
AcmeSIS	<code>SIF_Message[SIF_Event/SIF_Header/SIF_SourceId="AcmeLibrary"]//StudentPersonal/LocalId</code>

Table 3.5.3-1: XML Filter Example 1

In this example the whole SIF message is filtered if the destination agent is not the same as the `SIF_OriginalHeader/SIF_Header/SIF_SourceId` and if the `SIF_LogEntry` was published by another SIF agent in the zone.

Agent Id	SIF XML Filter
AcmeLibrary	<code>SIF_Message[/SIF_Message/SIF_Event/SIF_ObjectData/SIF_EventObject/SIF_LogEntry[@Source="Agent"]/SIF_OriginalHeader/SIF_Header[SIF_SourceId!="AcmeLibrary"]]</code>
AcmeTrans	<code>SIF_Message[/SIF_Message/SIF_Event/SIF_ObjectData/SIF_EventObject/SIF_LogEntry[@Source="Agent"]/SIF_OriginalHeader/SIF_Header[SIF_SourceId!="AcmeTrans"]]</code>
AcmeSIS	<code>SIF_Message[/SIF_Message/SIF_Event/SIF_ObjectData/SIF_EventObject/SIF_LogEntry[@Source="Agent"]/SIF_OriginalHeader/SIF_Header[SIF_SourceId!="AcmeSIS"]]</code>

Table 3.5.3-2: XML Filter Example *SIF_LogEntry*

3.5.3.1 SIF XML Filter Process Rules

When processing a SIF Message for an agent, if SIF XML filters have been defined for the recipient of a SIF message, the Zone Integration Server **MUST** be compliant with the following guidelines.

For each SIF XML filter that has been defined for the destination agent, the ZIS executes a filter against the message. For each match that is found in the message, the ZIS **MUST** remove each node. If the match that is executed results in a match of the document element, *SIF_Message*, the entire message has been held by the XML filter rule, and the message **MUST NOT** be delivered to the recipient. Otherwise, if XML validation is enabled, the ZIS **SHOULD** validate the message after applying all element level security rules and follow normal procedures if validation fails.

3.5.3.2 Implementation of SIF XML Filter Syntax

SIF XML filters **SHOULD** be implemented using support for XPath. The ZIS **MUST** also support the ability to add one or more SIF XML filters to an agent. Multiple SIF XML filter rules may be created in order to enforce a single security rule across all of the different message types that may contain the affected data elements. While a Zone Integration Server **SHOULD** support XPath rules and allow them to be edited by the end user, nothing within this specification prevents a Zone Integration Server from also presenting a more simplified interface to the end user, in which case, the ZIS itself may translate end user options to the associated XPath behind the scenes.

Implementation notes.

MUST NOT filter the SIF Object "root element" in *SIF_Events* to have the message removed. Target the document element *SIF_Message* to filter *SIF_Event* messages.

It is **NOT RECOMMENDED** to filter optional elements when agents require these elements to exist.

The ZIS **MUST NOT** repack *SIF_Response* streams if an object is filtered from the *SIF_Response* stream. If a response is empty after the filter has been applied the *SIF_Response* **SHOULD** still be delivered.

If the *SIF_Response* message is filtered the ZIS **MUST** implement the QoS implementation for when a *SIF_Response* packet is dropped by the Zone.

The implementation of the XPath **MAY** need to alter the XPath for namespace support.

3.5.4 Zone Context Registry

Zone Integration Servers **MUST** maintain a registry of the contexts used in each zone in order to perform contextual message routing and to populate the *SIF_ZoneStatus/SIF_Contexts* element. This registry will always contain, at a minimum, the official list of contexts defined by the version of SIF that the ZIS supports. Beginning with SIF Implementation Specification 2.0, each zone will, at a minimum, support the *SIF_Default* context. A ZIS **MAY** allow additional contexts to be defined within the context registry at the discretion of the ZIS administrator.

Access Control Lists within each context **MUST** be supported by the ZIS and be available for management by the ZIS administrator. The set of ACL permissions for an agent within a context **MUST** be independent of ACL permissions for that agent within a different context.

An agent can obtain a list of all contexts currently defined in a zone by requesting the *SIF_ZoneStatus* object and enumerating the children of its *SIF_Contexts* element. An agent can determine its ACL permissions within the zone and each context by referring to the *SIF_AgentACL* object and enumerating the permissions and contexts defined within it.

3.5.5 Administration

A ZIS **MUST** provide an interface for Zone Administrators to configure zone settings, including access control permissions. Given the distributed nature of SIF, it is **RECOMMENDED** this be a Web-based interface. Some of the areas that require administration are:

Administration

Start and stop the ZIS and/or set the state of the ZIS to "asleep" or "awake."

Security Policies

A ZIS must provide an interface for administering access control permissions as described above.

Administering the minimum `SIF_EncryptionLevel` for the zone (if only one encryption level is supported, configuration options are unnecessary).

Administering the minimum `SIF_AuthenticationLevel` for the zone (if only one authentication level is supported, configuration options are unnecessary).

It may also include installing client and server certificate administration.

Zone Settings

If the ZIS supports more than one SIF version it must support configuration of which SIF versions are used in a zone.

If the ZIS supports more than one transport protocol, it must allow for configuration of which transports agents can use to communicate, including limiting communication to SIF HTTPS.

The ZIS must support configuration of the minimum acceptable `SIF_MaxBufferSize` for the zone.

If message validation is supported and configurable, configuration to enable or disable message validation in a Zone should be available.

Logging

Capture error and message logs to aid in tracking pending, successful and failed delivery of messages.

Reporting

Report zone status and statistics.

Testing

Provide a mechanism to "ping" Push-Mode agents.

3.5.6 Support Selective Message Blocking (SMB) to Resolve Deadlocks

3.5.6.1 Description

Selective Message Blocking is a feature that **MUST** be implemented by a ZIS to enable non-multitasking agents, unable to persist portions of their message queue locally, to request information from other agents while processing a `SIF_Event` message, without causing communication "deadlock" between an agent and a ZIS.

This feature allows an agent to inform the ZIS with an "Intermediate" `SIF_Ack` message that the ZIS must temporarily stop delivering `SIF_Event` messages to the agent. The "Intermediate" `SIF_Ack` message must not be used by agents in response to messages other than `SIF_Event`. The ZIS, however, can deliver other agent-destined messages, `SIF_Request` and `SIF_Response`, to this agent. After it finishes processing the `SIF_Event` message this agent sends the "Final" `SIF_Ack` message to the ZIS, which will discard the blocked `SIF_Event` message and resume normal delivery of all messages, including `SIF_Events`. SMB is supported for both Push and Pull modes.

SMB will not be supported for any Zone Service messages. In particular, asynchronous Notification message packets will not be blocked.

3.5.6.2 Requirements

If, after attempting delivery of a `SIF_Event` message to an agent, the ZIS receives an Intermediate `SIF_Ack` (`SIF_Status/SIF_Code=2`) from the agent, the event is blocked and all `SIF_Event` messages destined for the agent, whether already in the queue or that arrive while blocked, are considered frozen. The ZIS will not deliver any `SIF_Event` messages that are frozen.

If no `SIF_Ack` at all is received, or if a transport error occurs, this `SIF_Event` must be considered an undelivered message. The next message to be delivered to the agent will be this event.

The ZIS must not deliver another `SIF_Event` message to the agent until a "Final" `SIF_Ack` is received (`SIF_Status/SIF_Code=3`), giving the ZIS permission to discard the original event and resume event delivery. The `SIF_OriginalMsgId` in the "Final" `SIF_Ack` **MUST** contain the `SIF_MsgId` of the blocked `SIF_Event`.

If `SIF_Events` are frozen, the next message to be delivered is the oldest message that is not a `SIF_Event` message. Once `SIF_Events` are unfrozen, all remaining messages in this agent's queue, including `SIF_Events`, will be delivered in the order in which they have been received by the ZIS.

If the ZIS receives a `SIF_Wakeup` or `SIF_Register` message then the block on any frozen `SIF_Event` messages will be removed and the originally blocked message will be the next message delivered to the agent.

3.5.6.3 Example

For a detailed example of SMB, see the

[Selective Message Blocking \(SMB\) Example](#).

3.5.7 Quality of Service Implementation

The Zone Integration Server is required to maintain a reliable list of all messages that support buffering. These message types include: `SIF_Response`, `SIF_ServiceInput`, `SIF_ServiceOutput`, and `SIF_ServiceNotify` messages in order to satisfy the Quality of Service validations that are present for these messages. These messages are collectively identified as the buffered message types below.

Once a buffered message stream has been completed, either by receiving the last packet or by failing one of the validations applied, knowledge of this buffered message stream will no longer need to be maintained by the ZIS. If the agent initiating the buffered message stream attempts to send any more messages and message stream has been completed, the messages will automatically fail.

If the message stream terminated because of an error, and the ZIS has initiated or has been made aware of the error, notification of the failed message stream will be sent to the destination agent.

There remain three cases where a destination agent will not receive a complete Buffered Message stream for request/response message types.

1. The responding agent never replies.
2. The agent starts a buffered stream, but never finishes the buffered stream by sending a message with the `SIF_MorePackets` element set to "No".
3. The agent attempts a buffered stream, but the ZIS is unable to parse the message enough to read the `SIF_ServiceMsgId` or `SIF_RequestMsgId`. If this case occurs, and the responding agent sends a subsequent buffered message that is readable, the destination agent will be notified of the problem because subsequent packets will not pass the `SIF_PacketNumber` validation.

Management of the message buffer tracking cache maintained by the ZIS is left up to the ZIS implementation. The ZIS is required to maintain the cache for an amount of time configurable by the ZIS administrator.

If a ZIS does remove tracking information for a message, it **MUST** notify any agents that have received packets from the buffered message stream.

When an open message buffer cache entry is removed by the administrator or a timeout of the record, the ZIS **SHOULD** publish a `SIF_LogEntry` and a `SIF_Error` indicating the reason it was removed.

3.6 Message Processing

To ensure interoperability, SIF defines a set of messages that are exchanged between agents and Zone Integration Servers. The SIF messages are used to perform various operations such as provision, subscription, event reporting, request and response, and ZIS administration.

3.6.1 Message Validation

SIF recommends that each message receiver validate any incoming message to ensure that it is a valid SIF message. A message receiver should discard any messages that do not conform to the definition of `SIF_Message` and return an error to the originator of the message.

This specification will evolve over time to include new messages and modifications to messages that have been defined. Each agent and ZIS should explicitly define which version(s) of the specification they support and validate each incoming message according to its version.

The SIF Association provides an XML Schema

[[SCHEMA](#)] corresponding to this version of the specification for ZIS and agent implementations that choose to perform optional message validation. Implementations are free to include additional validation above and beyond the validation capabilities that XML Schema provides.

The schemas for all versions of this specification are available from the SIF Association and can be referenced by ZIS and agent implementations that choose to perform optional message validation. This allows implementations to choose schemas based on the particular version in use by an agent or a ZIS. SIF messages **MUST NOT** be transmitted with hard-coded references to DTDs, schemas or other validation mechanisms. The XML "doctype decl" (`<!DOCTYPE SIF_Message...`) **MUST NOT** occur in SIF XML messages, nor should `xsi:schemaLocation` be used on `SIF_Message`.

The schemas for supported versions of the SIF Implementation Specification enforce ordering of elements and data typing within objects, as per the element tables given in [Infrastructure](#) and [Data Model](#). In the event that ZIS and agent implementations choose not to perform message validation, ZIS and agent implementations must still send elements as ordered with the types specified in the element tables (i.e., well-formed AND valid XML must be sent for approved objects even if validation is known to be turned off). When XML validation is turned off, the sending of draft and locally-defined objects not included in the schemas becomes possible, and these objects may experimentally be sent as desired until they make their way into future versions of the specification and supporting schemas.

ZIS implementations are in the unique position of not only sending messages they themselves formulate; they also forward messages received from agents. When optional message validation is not being performed by a ZIS, it is possible that the ZIS may receive a well-formed but invalid XML message from a non-compliant agent. Under these circumstances, and being the routing mechanism it is, a ZIS is under no obligation to correct an invalid XML message it receives from an agent for delivery to other agents. Zone administrators can prevent invalid XML messages from being delivered if the ZIS supports message validation. Should it receive an invalid but well-formed message from an agent, a ZIS not performing message validation delivers the message like any other to destination agents.

3.6.2 Message Identification

Each message originating from an agent or ZIS **MUST** have a unique message identifier (`SIF_MsgId`). In order to eliminate the possibility of duplicated message identifiers, and to provide a consistent way of generating these identifiers, SIF requires the use of a globally unique identifier

[[RFC 4122](#)] as message identifiers.

The reason that a unique identifier is required is that many messages are handled asynchronously in SIF. This means, for example, that `SIF_Responses` for a given `SIF_Request` message may not arrive until sometime in the future. When the `SIF_Response` arrives, it will contain the original `SIF_MsgId` but no other information about the original message is guaranteed to be provided. The message originator must ensure that it will be able to match up the `SIF_Response` with the original message based solely on the message identifier.

For further information concerning the generation of GUIDs, see [[RFC 4122](#)].

3.6.3 Message Security

Because of policy or legislation, providers of extremely sensitive data must never expose that data over an insecure channel. An insecure channel at delivery time is one whose levels of authentication (`SIF_AuthenticationLevel`) and data encryption (`SIF_EncryptionLevel`) fall below the values specified by the originating sender. Once the data is communicated to the ZIS, the originator of the message depends upon the ZIS to enforce the security levels requested and the ZIS must not deliver that message to recipient agents using an insecure channel. The originating agent requests the use of a secure channel at delivery time by incorporating a `SIF_Security` element in the header of the message. The `SIF_Security` element contains `SIF_AuthenticationLevel` and `SIF_EncryptionLevel` elements that define the minimum level of security a data transport channel must provide upon delivery. If a ZIS does not deliver a message due to insufficient security of the connection with a recipient agent, it is recommended that the ZIS log the inability to deliver the message due to security requirements.

The only SIF messages that currently originate with an agent and that are ultimately delivered to other agents are `SIF_Request`, `SIF_Response` and `SIF_Event`. An originating agent may add a `SIF_Security` element to all messages, but these three messages are the only ones where `SIF_Security` will be examined and processed by the ZIS. `SIF_Security` is used by an originating agent to specify the security requirements of the communication channel between the ZIS and any recipient agent at delivery time. The semantics of including the `SIF_Security` element on messages other than `SIF_Request`, `SIF_Response` and `SIF_Event` are reserved for future versions of the specification.

The specification provides several levels of authentication and encryption protection.

3.6.3.1 SIF_AuthenticationLevel

0

No authentication required and a valid certificate does not need to be presented.

1

A valid certificate must be presented.

2

A valid certificate from a trusted certificate authority must be presented.

3

A valid certificate from a trusted certificate authority must be presented and the CN field of the certificate's Subject entry must match the host sending the certificate.

The CN field is more commonly known as the "Common Name" field. `SIF_AuthenticationLevel 3` requires that the CN contents match the host where the message was originated. For instance, a CN entry could be "sifinfo.org" or perhaps "207.95.37.30". If a ZIS at SifInfo.org (IP address 207.95.37.30) contacts an agent at MyAgent.sifinfo.org, the agent's SIF HTTPS transport layer can look at the CN entry in the certificate that was presented by the ZIS and compare it to the actual IP address of the ZIS.

`SIF_AuthenticationLevel 3` ensures that not only a valid and trusted certificate was presented but that the agent is actually communicating to the ZIS located at the IP address referenced in the certificate.

Because security is a cornerstone of the SIF specification, it is recommended that all ZIS and agent implementations support client authentication as well as server authentication. When client authentication is being used, the connection first authenticates the server (the party that is being contacted) and if the authentication was successful, the server will request that the client present its certificate for authentication. In this manner, both the ZIS and the agent confirm that they are communicating with the proper parties.

Since client authentication is not universally available in all SIF HTTPS implementations, client authentication is only recommended. The need for client authentication is reduced somewhat by using asynchronous message delivery (Push mode) since the ZIS and the agent are both server type applications and will authenticate each other. The need for client authentication is greater for those agents polling for messages (Pull mode) because the ZIS never has to initiate contact with the agent.

3.6.3.2 SIF_EncryptionLevel

0

No encryption required

1

Symmetric key length of at least 40 bits is to be used

2

Symmetric key length of at least 56 bits is to be used

3

Symmetric key length of at least 80 bits is to be used

4

Symmetric key length of at least 128 bits is to be used

If a `SIF_Request`, `SIF_Response` or `SIF_Event` is received by the ZIS that does not contain a `SIF_Security` element, the ZIS assigns the lowest level (0) to both the `SIF_AuthenticationLevel` and `SIF_EncryptionLevel`, unless a Zone administrator has configured higher minimum encryption and authentication levels for the Zone. This means that the ZIS may distribute this message to any agent that has registered with the ZIS subject to the access control security provisions in place for the zone.

The lack of a `SIF_Security` element does not mean that the message will be transported in an insecure manner. Recipient agents communicating with the ZIS over secure channels will receive the message in a secure manner, consistent with the connection. Omitting the `SIF_Security` element simply allows for those agents that communicate over insecure channels to receive the message, should a zone allow for insecure channels. A zone administrator can prevent messages without `SIF_Security` elements being communicated over insecure channels by configuring the ZIS and agents in the zone such that a minimum security level is maintained, below which communication is impossible.

For ZIS and agent implementations that support communication protocols or transport implementations where the security of a channel cannot be determined at delivery time, it is recommended that the zone administrator configure the ZIS and agents in the zone such that a minimum security level is maintained, below which insecure connections cannot be established.

3.6.3.3 Notes on SIF_AuthenticationLevel

If authentication based on certificates is being used, care needs to be given to determine if Level 2 (anonymous certificates) will provide the necessary level of protection. With Level 2 authentication, it is possible to use a web browser to make secure connections to the ZIS using the certificates that are built into the browser. This level of authentication is what is used by almost all Internet transactions (stock trading, shopping, financial, etc.). Level 2 does expose the user to a risk of a "man-in-the-middle" attack that can't occur using Level 3 authentication.

Level 3 mandates that a certificate issued by a trusted authority, (i.e. school district), be installed in the web browser before the browser will be able to connect to the ZIS. This may place unnecessary burdens on the client especially if it is likely that authorized users may wish to connect to the ZIS using a variety of browsers.

3.6.3.4 Notes on SIF_EncryptionLevel

The major governing factor as to the strength of data encryption is the length of the cipher key. A 128-bit implementation typically provides stronger encryption than an 80-bit implementation. Please note that support of some SIF encryption levels may be subject to export control, limiting distribution of all levels in all countries

[EXPORT].

There are also two main types of cipher algorithms. The first is called a symmetric cipher, which uses the same key to encrypt and decrypt the data. The second type is called public-key cipher, which depends upon using a private key of the sender along with the public key of the receiver. Because of the nature of public-key ciphers, a larger number of bits must be used to achieve a comparable level of encryption strength.

The `SIF_EncryptionLevel` bit sizes are based on symmetric ciphers. A table that lists the equivalent key length for a public-key cipher is listed below.

Symmetric Key Length	Public Key Length	Strength
40 bits	256 bits	Very weak, not recommended except for very minimal protection (i.e. prevents casual snooping but can be broken in minutes by knowledgeable attackers).
64 bits	512 bits	Weak. The current U.S. "standard" has been bumped up to 64 bits from 56 bits but the key length is still weak for sensitive data.
80 bits	768 bits	Moderate
128 bits	2048 bits	Strong, recommended for Internet

Table 3.6.3.4-1: Key Lengths

For more information regarding this topic, please refer to Chapter 7 of

[Schneier].

3.6.4 Message Robustness

It is important for SIF to guarantee message delivery no matter what happens during delivery of a message, including an unexpected network breakdown or system crash. This requires that each agent and ZIS save each message in permanent storage. At delivery time it is possible, however, for a ZIS to be prohibited from delivering a message due to security requirements requested by originating agents for individual messages. If this occurs, it is recommended that ZIS implementations discard the affected message so that delivery of other messages may proceed. If the ZIS does discard the message, the ZIS **MUST** report a `SIF_LogEntry` event with the appropriate error category and code, containing a copy of the `SIF_Header` from the original message. In addition, it is recommended that the ZIS log the delivery failure to its own log.

When a message is delivered under normal circumstances by a ZIS, an agent will return an "Immediate" `SIF_Ack` or a `SIF_Ack` with any applicable error condition, signaling the ZIS that it may delete the message from permanent storage. In the case of events, agents may also return an "Intermediate" `SIF_Ack` to invoke Selective Message Blocking (SMB). In that case, the ZIS will not delete the current `SIF_Event` from permanent storage until the agent sends a "Final" `SIF_Ack` to the ZIS.

When a message is sent to the ZIS under normal circumstances by an agent, the ZIS returns a successful `SIF_Ack` or a `SIF_Ack` with any applicable error condition to indicate to the agent that it has in fact received the message and that the agent may delete the message from any permanent storage.

If a ZIS or agent encounters a transport error in sending a message, it is recommended that the sender retry sending the message. Transport errors where retrying the message is warranted include, but are not limited to, a connection close without a `SIF_Ack` returned, a transport error or a `SIF_Ack` with an error category of 10 indicating a connection cannot currently be established, etc. A ZIS in particular must retry delivery of messages from the agent queue until a `SIF_Ack` that removes the message from the agent's queue is received, subject to certain undeliverable error conditions (e.g. security requirements cannot be negotiated, maximum buffer size too small, etc.). Facing such error conditions, other potentially irresolvable transport errors, or if a `SIF_Ack` is returned with any other type of error category, the sender may decide not to retry or—when queued, to delete—a message to avoid a potential deadlock condition. Agents returning `SIF_Ack` messages with error conditions should be aware that such acknowledgements will remove the currently pending message from their delivery queue.

3.6.5 Message Cycle

All SIF messages follow the same model. The sender posts a message and receives a `SIF_Ack` back as a response. The posting of the message by the sender and the receipt of the `SIF_Ack` from the receiver constitutes one complete cycle. Agents and ZISes can function as senders or receivers, depending on the type of message. The message process is identical, regardless of the type of message being sent.

If for any reason a sender inadvertently resends a message with a given `SIF_MsgId` and the receiver detects this, the receiver may return a `SIF_Status` code indicating that it already has the message. This `SIF_Status` code is considered a success; the receiver simply discards the duplicate message and continues handling of the original message.

3.6.6 Message Delivery

There are two models for delivering messages to an agent, "Push" and "Pull." An agent specifies which mode it wants to use when it registers with the ZIS.

"Push" refers to the action by a ZIS to actively deliver messages to an agent without the agent having to initiate contact with the ZIS. When the ZIS receives a message for an agent and the agent is not in "Sleep" mode; the ZIS will initiate contact with the agent and send the message to the agent.

"Pull" refers to the action by an agent to explicitly request a single message from the ZIS. When an agent is ready to receive a message, it sends a "Pull" request to the ZIS, to obtain a message that the ZIS has saved in the queue for the agent. After receiving the pull request, the ZIS will examine the agent's queue and either returns a message or a status code indicating that no messages are available for the agent.

Both modes serve useful purposes. The key requirement is that both an agent and its ZIS must communicate using the SAME mode to avoid potential conflicts.

At delivery time, be it in push or pull mode, a ZIS may encounter messages that it is prohibited from delivering, e.g. due to security requirements requested by originating agents for individual messages, etc. If this occurs, it is recommended that ZIS implementations discard the affected message(s) so that delivery of other messages may proceed. If the ZIS does discard a message, the ZIS **MUST** report a `SIF_LogEntry` event with the appropriate error category and code, containing a copy of the `SIF_Header` from the original message. `SIF_LogEntry/SIF_Desc` must contain the `SourceId` of the agent that has failed to receive the message. In addition, it is recommended that the ZIS log the delivery failure to its own log.

3.6.6.1 The "Push" Model

When an agent has registered using the "Push" mode, the agent assumes that the ZIS will open a transport connection and send the next available message to the agent. An agent can reply to the sent message with an "Immediate" or optionally—in the case of `SIF_Events`—an "Intermediate" `SIF_Ack`, invoking Selective Message Blocking (SMB); it can also reply using a `SIF_Ack` with any applicable error condition. "Immediate" or error `SIF_Acks` remove the current message from the agent's queue, freeing any remaining or future messages to be delivered to the agent. A "Final" `SIF_Ack` sent to the ZIS will terminate SMB, removing the frozen event from the agent's queue, freeing any remaining or future messages to be delivered to the agent.

3.6.6.2 The "Pull" Model

When an agent has registered using the "Pull" mode, the agent requests a message from the ZIS by sending a `SIF_GetMessage` message to the ZIS.

An agent can only issue a `SIF_GetMessage` to request a message if the agent has previously sent a successful `SIF_Register` message specifying Pull mode. If the ZIS receives a `SIF_GetMessage` request and the agent hasn't registered using the Pull mode, the ZIS must return a `SIF_Ack` containing an error category of Registration and an error code indicating that the agent has registered using Push mode.

After receiving a `SIF_GetMessage` request from an agent, the ZIS will return the next message available for delivery to the agent, subject to Selective Message Blocking. The criteria used to select the message are identical to that used if the ZIS were to Push a message to an agent.

If a message is available for the agent, the ZIS will return a `SIF_Ack` message with a `SIF_Status/SIF_Code` of 0 and `SIF_Status/SIF_Data` containing the message from the queue:

```
<SIF_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>ABCD1058E028D076F083738296372D4E</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>SifInfo_TestZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>1058ABCDE028D076F083283BC63E6276</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>0</SIF_Code>
    <SIF_Data>
      <SIF_Message Version="2.4">
        <SIF_Event>
          <SIF_Header>
            <SIF_MsgId>AB34DC093261545A31905937B265CE01</SIF_MsgId>
            <SIF_Timestamp>2006-02-18T08:39:12-08:00</SIF_Timestamp>
            <SIF_SourceId>RamseyLib</SIF_SourceId>
          </SIF_Header>
          <SIF_ObjectData>
            <SIF_EventObject ObjectName="StudentPersonal" Action="Change">
              <StudentPersonal RefId="D3E34B359D75101A8C3D00AA001A1652">
                <Name Type="04">
                  <FirstName>William</FirstName>
                </Name>
              </StudentPersonal>
            </SIF_EventObject>
          </SIF_ObjectData>
        </SIF_Event>
      </SIF_Message>
    </SIF_Data>
  </SIF_Ack>
</SIF_Message>
```

Example 3.6.6.2-1: The "Pull" Model - `SIF_Status/SIF_Code` of 0

A pull-mode agent removes the returned message from its queue in one of three ways. In each case the value for the `SIF_OriginalMsgId` element in any `SIF_Ack(s)` created by the agent originates from the `SIF_MsgId` of the `SIF_Message` returned as `SIF_Data` by the ZIS. Typically a pull-mode agent removes the message from its queue by sending an "Immediate" `SIF_Ack` to the ZIS; an agent may also send a `SIF_Ack` with any applicable error condition to the ZIS. The ZIS then removes the message from the agent's queue and returns a successful `SIF_Ack`. If the message is a `SIF_Event` and the agent wishes to invoke SMB, it can instead notify the ZIS that it is processing the event by sending an "Intermediate" `SIF_Ack` (which the ZIS acknowledges with a successful `SIF_Ack`) and later sending a "Final" `SIF_Ack` when the `SIF_Event` processing is complete. When the ZIS receives the "Final" `SIF_Ack`, it removes the `SIF_Event` from the agent's queue and returns a successful `SIF_Ack`.

If there are no messages in the agent's queue that can be delivered, the ZIS will return a `SIF_Ack` message with a `SIF_Status/SIF_Code` of 9 to indicate that there are no messages available for the agent:

```
<SIF_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>ABCD1058E028D076F0835E32AC89E048</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>SifInfo_TestZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>1058ABCDE028D076F08365109BE7C892</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>9</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example 3.6.6.2-2: The "Pull" Model - `SIF_Status/SIF_Code` of 9

3.6.6.2.1 Version Management

When a pull-mode agent supports multiple SIF specification versions, the version of the `SIF_Ack` message returned by the ZIS must match the version of any `SIF_Message` contained in `SIF_Status/SIF_Data`. For example, if an agent supports versions 1.1 and 1.5 (or 1.*) and the next message in the agent's queue has a `SIF_Message/@Version` value of 1.5, the `Version` attribute of the `SIF_Ack` message returned by the ZIS must be 1.5, even if the pull-mode agent sent its `SIF_GetMessage` in a 1.1 `SIF_Message`. For an agent that supports both 1.1 or later versions and pre-1.1 version(s) (e.g. 1.0r2), when the next message in the agent's queue is from a pre-1.1 agent, the ZIS must return the message in a `SIF_Ack` message as defined by the pre-1.1 specification.

3.6.6.3 "Multiple Version" Zones

It is possible for a zone to contain agents written to different versions of the SIF Implementation Specification. This is true if a ZIS supports multiple versions in a zone and has at least one version in common with all registered agents. It is possible that two agents in the same zone—both successfully registered—have no version in common, and this affects message delivery by the ZIS in the following manner.

When the next message to be delivered to a given agent has a `SIF_Message/@Version` attribute that the agent is known not to support, the ZIS cannot successfully deliver that message to the agent without conversion. Should a ZIS implementation choose to convert messages on the fly as a "value-add" feature, it is free to do so; this specification does not prescribe how to convert messages, and support for such conversion is implementation-dependent. However, if the ZIS does not or cannot convert the message such that it can be delivered, it should discard the pending message so that delivery of other messages may proceed. If the ZIS does discard the message, the ZIS **MUST** report a `SIF_LogEntry` event with the appropriate error category and code, containing a copy of the `SIF_Header` from the original message. `SIF_LogEntry/SIF_Desc` must contain the `SourceId` of the agent that has failed to receive the message. In addition, it is recommended that the ZIS log the delivery failure to its own log.

3.7 Infrastructure Transport Layer

The Infrastructure messages are used by SIF to encapsulate and transfer the data objects. They form a messaging Application Program Interface (API) expressed in XML.

The entire SIF Infrastructure API is expressed in XML and does not have dependencies upon any underlying transport layer to provide functionality other than the movement of the XML from client to server and back. This ensures that infrastructure messages can be carried over a variety of communication transports.

The infrastructure depends upon the transport layer to provide a reliable connection to move messages back and forth from client and server. The transport layer is also responsible for providing transport-level security by means of data encryption and authentication. Some transport layers even provide data compression, which can be an important factor when processing a large volume of XML messages.

By delegating the authentication, compression, and encryption to the transport layer, it makes the user interface to the transport simpler. A client that wishes to send an infrastructure message assembles the message in XML and then hands it off to the transport layer for delivery. The transport layer takes the XML message and transfers it to the server where it is taken from the transport layer and processed.

In moving from the client to the server, the transport may have compressed, encrypted, and authenticated the connections but all of this is transparent to the users of the Infrastructure API. To the user, it is XML in and XML out.

Different types of transports are or will become available providing various features and benefits. An agent or ZIS **MAY** employ multiple transport protocols but they **MUST** support SIF HTTPS.

Please note that throughout this specification transport layer errors are sometimes illustrated as `SIF_Ack` messages with `SIF_Error/SIF_Category` of Transport and applicable error codes. Under many transport error conditions, these `SIF_Ack` messages could not be returned or sent by the remote host. Depending on the SIF infrastructure transport layer implementation these messages may be generated by the implementation (e.g. when a connection to a server cannot be established), or may occur as transport layer errors or exceptions in the underlying network operating system or transport protocol. Both should be treated equivalently.

3.7.1 SIF HTTPS Transport

In order to ensure that agents and Zone Integration Servers can communicate with each other regardless of vendor or platform, all agent and ZIS implementations **MUST** support the SIF HTTPS transport layer protocol.

SIF HTTPS is a combination of the HTTP 1.1 protocol

[RFC 2616] with secure socket layer (SSL) protocols, resulting in an easy-to-use and secure transport protocol. The **RECOMMENDED** SSL implementation is TLS 1.0 [RFC 2246]; however, SSL 3.0 [SSL3] is also supported and SSL 2.0 client hellos [SSL2] used to negotiate TLS 1.0 or SSL 3.0 connections are also permitted. Support for the SSL 2.0 protocol itself—aside from its client hello message—is not provided in SIF. Due to the age of the SSL 3.0 and SSL 2.0 protocols and the increasing prevalence of TLS 1.0, The SIF Association expects to deprecate support for the SSL 3.0 protocol and SSL 2.0 client hellos in future major releases of this specification.

Being based upon HTTP 1.1, the SIF HTTPS and SIF HTTP protocols support persistent or keep-alive connections that greatly increase the message throughput between sender and receiver. This is an especially important factor when using HTTP in conjunction with secure socket layers, where there is a significant amount of overhead when initially opening a connection.

When using HTTP 1.1 with SIF, [RFC 2616] can be used as a reference, however SIF uses a subset of the HTTP 1.1 protocol. For example, only the POST method and the 200-OK response notice are used by the SIF HTTPS protocol.

Support of Transfer Encoding and data chunking ([RFC 2616], Section 3.6) is not required for SIF HTTPS. An implementation of the protocol may support Transfer Encoding and data chunking but it must be able to communicate successfully with a client or server that does not support this feature.

Because protocol changes are handled at the Infrastructure XML API level, a client or server must not use the `Connection: Upgrade` or `Upgrade: xxx` headers to invoke a request for a protocol change. If a client or server receives an upgrade header, it must ignore that header and not change communication protocols.

3.7.1.1 HTTPS Request/Response Model

A client is the party (agent or ZIS) who initiates a connection to a remote machine. The remote end (ZIS or Push-Mode Agent) is known as the server.

A client using the SIF HTTPS protocol opens a connection to the server and sends a HTTP 1.1 POST request with the SIF Infrastructure XML message as the POST payload. The server responds with an HTTP response with the Infrastructure XML acknowledgement message as the response payload.

The default behavior for HTTP 1.1 is to use persistent or "keep-alive" connections. When operating in this mode, the client may send additional POST requests and receive the HTTP responses using the same connection. Clients **SHOULD** use persistent connections for performance reasons but **MUST** be able to use non-persistent connections if the server does not wish to use persistent connections.

3.7.1.2 UTF-8 Encoding

Clients **MUST** encode the XML message using UTF-8; servers **MUST** be able to process UTF-8-encoded XML and **SHOULD** expect all incoming SIF XML messages to be encoded using UTF-8.

3.7.1.3 HTTP Request Headers

The following HTTP request and common headers defined in

[RFC 2616] **MUST** be present in all SIF HTTPS messages sent by a client:

Header	Description	Required Contents
Content-Length	The exact size of the attached payload (XML message)	
Content-Type	Describes the contents of the request. Firewall and web server programs can filter messages going through a network by examining this header.	application/xml;charset="utf-8"
Host	Specifies the Internet host and port number of the destination server	

Table 3.7.1.3-1: HTTP Request Headers

Note that all header values **MUST** conform to the requirements of

[RFC 2616] and **MAY** take equivalent forms subject to those requirements (e.g. `application/xml;charset=utf-8` (no quotes), `application/xml; charset=utf-8` (optional spacing), etc.).

In addition to the headers above, a client may include a `Connection: close` header in the HTTP request if it wishes to close the current connection after receiving the response. If this header is included, the client **MUST NOT** send additional requests on this connection. The client **MUST** close the connection after receiving the response.

Clients may also include an "Expect: 100-continue" header (see below).

Additional headers beyond the required and optional headers listed here **MAY** be included by a client; however, the server **MUST** be able to successfully process POST requests that only contain the required headers.

```
POST /MyPath HTTP/1.1
Content-Length: 420
Content-Type: application/xml;charset="utf-8"
Host: sifinfo.org:8000

<SIF_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>56409F0C01FBD1C44300B4518E100765</SIF_MsgId>
      <SIF_Timestamp>2006-04-11T18:18:13-05:00</SIF_Timestamp>
      <SIF_SourceId>SifInfo_TestAgent</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_Ping />
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>
```

Example 3.7.1.3-1: SIF HTTPS Request

Implementations of SIF HTTPS **MUST** be able to specify the value for the path (`/MyPath` in the example) as the agent or ZIS may require a specific value for routing purposes.

3.7.1.4 HTTP Response Headers

The following HTTP response and common headers defined in [RFC 2616] must be present in all SIF HTTPS responses messages sent by a server:

Header	Description	Required Contents
Content-Length	The exact size of the attached payload (XML message)	
Content-Type	Describes the contents of the request. Firewall and web server programs can filter messages going through a network by examining this header.	application/xml;charset="utf-8"
Date	The current date and time in the format described in RFC 2616 Section 3.3. Note that the date is UTC based and NOT local time.	
Server	Identifies the server sending the response. Clients may use this information to infer information about the server being contacted (vendor, model, version, capabilities, etc.)	

Table 3.7.1.4-1: HTTP Response Headers

Note that all header values **MUST** conform to the requirements of

[RFC 2616] and **MAY** take equivalent forms subject to those requirements (e.g. `application/xml;charset=utf-8`, `application/xml; charset=utf-8`, etc.).

In addition to the headers above, a server **MAY** include a `Connection: close` header in the HTTP response if it wishes to close the current connection after sending the response. The server **MUST** close the connection after sending the response.

The server **MAY** include additional headers; however, the client **MUST** be able to successfully process response notices that only contain the required headers and optional header listed here.


```

HTTP/1.1 200 OK
Content-Length: 529
Content-Type: application/xml;charset="utf-8"
Date: Mon, 02 Apr 2001 23:32:00 GMT
Server: SIFZIS;V1.1

<SIF_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>4A900E10F4E675CF4A01B4518E100765</SIF_MsgId>
      <SIF_Timestamp>2006-04-11T18:18:13-05:00</SIF_Timestamp>
      <SIF_SourceId>SifInfo_TestZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>SifInfo_TestAgent</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>56409F0C01FBD1C44300B4518E100765</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>0</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>

```

Example 3.7.1.4-1: SIF HTTPS Response

Although the SIF HTTPS protocol uses the 200-OK response notice to communicate all responses, agent or ZIS implementations could be built using existing web server infrastructures. As such, SIF HTTPS implementations should expect the possible receipt of other HTTP 1.1 response notices.

3.7.1.5 100 (Continue)

This response message status is generally returned if the client has included an `Expect: 100-continue` header in its request. Certain web server implementations return a 100 (Continue) status response even though the original request did not contain an `Expect: 100-continue` header. When a client receives an unexpected response with a 100 (Continue) status, it must discard that response and wait for a subsequent final (e.g. 200-OK) response. Clients explicitly requesting a 100 (Continue) status response by including an `Expect: 100-continue` header in a request should proceed with the request body according to section 8.2.3 of the HTTP 1.1

[RFC 2616] specification upon receipt of the 100 (Continue) status response.

A SIF HTTPS client may include an `Expect: 100-continue` header but generally does not. If it does, however, servers (ZIS and push-mode agent implementations) must handle the header according to section 8.2.3 of the HTTP 1.1 [RFC 2616] specification, possibly returning an intermediate response with 100 (Continue) status, for communication to proceed correctly.

3.7.1.6 3XX, 4XX, 5XX Notices

A server should return only 200-OK response notices. Servers built using existing web server technology might return other types of response notices. If a client receives any 3xx, 4xx, or 5xx response notices, it must treat these responses as if a transport error has occurred.

3.7.2 SIF HTTP Transport

The SIF HTTP protocol is identical to the SIF HTTPS transport without a secure socket layer to provide data encryption and authentication.

An agent or ZIS *MAY* implement the SIF HTTP transport but *MUST* implement the SIF HTTPS protocol.

Because of the sensitive data being exchanged in SIF, it is *RECOMMENDED* that only SIF HTTPS be used.

3.7.3 SIF HTTP(S) Transport Compression

It is possible that compression can improve network throughput in SIF implementations where large amounts of data are transferred over SIF HTTP(S), either horizontally or vertically. The HTTP 1.1 specification

[RFC 2616] allows for negotiating the content encoding (and compression) of server responses using the `Accept-Encoding` request header and the `Content-Encoding` response header. Registered content encodings include in addition to the default uncompressed identity encoding a number of compressed encodings: `gzip`, `compress` and `deflate`. A client can specify one or more encodings to use in a response along with its preference for each using `Accept-Encoding`, and the server responds accordingly, per the HTTP

specification. If the server does not support a requested encoding, it is recommended the server return a 406 (Not Acceptable) status code.

```
POST /MyPath HTTP/1.1
Content-Length: 420
Content-Type: application/xml;charset="utf-8"
Accept-Encoding: gzip
Host: sifinfo.org:8000

<SIF_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">
...
</SIF_Message>
```

Example 3.7.3-1: SIF client requesting compression of response

```
HTTP/1.1 200 OK
Content-Length: 24
Content-Type: application/xml;charset="utf-8"
Content-Encoding: gzip
Date: Wed, 25 Apr 2007 23:32:00 GMT
Server: SIFZIS

...compressed SIF_Ack...
```

Example 3.7.3-2: SIF server returning compressed SIF_Ack

The content encoding of any HTTP entity body, either in a request or a response, is indicated using the `Content-Encoding` header, which is considered a modifier to the `Content-Type` header. A client may compress or apply an encoding to the body of an HTTP request and indicate it has done so with an appropriate `Content-Encoding` value. It is recommended that a server that cannot or will not accept a particular encoding return a 415 (Unsupported Media Type) status code.

```
POST /MyPath HTTP/1.1
Content-Length: 149
Content-Type: application/xml;charset="utf-8"
Content-Encoding: gzip
Host: sifinfo.org:8000

...compressed SIF_Message...
```

Example 3.7.3-3: SIF client sending compressed SIF_Message

```
POST /MyPath HTTP/1.1
Content-Length: 149
Content-Type: application/xml;charset="utf-8"
Content-Encoding: gzip
Accept-Encoding: gzip
Host: sifinfo.org:8000

...compressed SIF_Message...
```

Example 3.7.3-4: SIF client sending compressed SIF_Message and requesting compression of response

With these HTTP-defined headers, SIF agents and Zone Integration Servers have the ability to compress or negotiate compression of SIF HTTP(S) request and response entity bodies using any version of SIF where the transport protocol is SIF HTTPS or SIF HTTP. However, to increase interoperability of agents and Zone Integration Servers that wish to compress requests or receive compressed responses beyond the level of trial and error in an environment where server status codes are not guaranteed, the following mechanisms were developed in SIF Implementation Specification Version 2.1.

3.7.4 SIF_Protocol/SIF_Property Accept-Encoding

In both `SIF_Register` and `SIF_ZoneStatus` the following `SIF_Property` is defined when used in conjunction with a `SIF_Protocol/@Type` value of `HTTPS` or `HTTP`:

SIF_Name	SIF_Value
Accept-Encoding	An Accept-Encoding header value as per HTTP 1.1 [RFC 2616].

This property indicates that an HTTP(S) server can accept corresponding content encodings with an appropriate Content-Encoding header value.

```
<SIF_Protocol Type="HTTPS" Secure="Yes">
  <SIF_URL>https://www.sifinfo.org/sifagent/MyAgent/</SIF_URL>
  <SIF_Property>
    <SIF_Name>Accept-Encoding</SIF_Name>
    <SIF_Value>gzip</SIF_Value>
  </SIF_Property>
</SIF_Protocol>
```

Example 3.7.4-1: SIF_Protocol with Accept-Encoding indicating acceptance of gzip (and identity)

```
<SIF_Protocol Type="HTTPS" Secure="Yes">
  <SIF_URL>https://www.sifinfo.org/sifagent/MyAgent/</SIF_URL>
  <SIF_Property>
    <SIF_Name>Accept-Encoding</SIF_Name>
    <SIF_Value>gzip;q=1.0, identity;q=0.5, *,q=0</SIF_Value>
  </SIF_Property>
</SIF_Protocol>
```

Example 3.7.4-2: SIF_Protocol with Accept-Encoding indicating no acceptance of encodings other than gzip or identity, gzip preferred over identity

The recommended compression algorithm for use in SIF is gzip. It is **NOT RECOMMENDED** that the identity (uncompressed) encoding ever be explicitly excluded in the Accept-Encoding SIF_Property.

3.7.5 HTTP Client Requirements

A client (ZIS, Push- or Pull-Mode Agent) that wishes to receive a compressed response **MUST** include an Accept-Encoding header, per HTTP 1.1, and **MUST** be prepared to handle a 406 (Not Acceptable) or other HTTP error, in which case the client **SHOULD** assume compression using the specified algorithm(s) is not supported and retry communication as per

SIF HTTPS Transport or SIF HTTP Transport above. Clients **MUST** be prepared to receive identity-encoded (unencoded) responses unless the client explicitly excludes identity in its Accept-Encoding header, which is **NOT RECOMMENDED**.

Zone Integration Servers **MAY** consult a Push-Mode Agent's registered SIF_Protocol/SIF_Property value where SIF_Name is Accept-Encoding before contacting the agent and **SHOULD** assume that posting a corresponding encoded entity body accompanied by the applicable Content-Encoding header value will be processed without content encoding support errors by the agent.

Push- and Pull-Mode Agents **MAY** consult a Zone's supported compression algorithms in the SIF_ZoneStatus/SIF_SupportedProtocols/SIF_Protocol/SIF_Property entitled Accept-Encoding in SIF_Name before contacting the Zone Integration Server and **SHOULD** assume that posting a corresponding encoded entity body accompanied by the applicable Content-Encoding header value will be processed without content encoding support errors by the ZIS.

3.7.6 HTTP Server Requirements

A server (ZIS or Push-Mode Agent) that receives an HTTP request with an Accept-Encoding header **MUST** process the request per HTTP 1.1's Accept-Encoding specification. It is **RECOMMENDED** that servers return a 406 (Not Acceptable) status when a requested encoding cannot be negotiated.

A server that receives an HTTP request with a Content-Encoding header specified **MUST** process the request per HTTP 1.1's Content-Encoding specification. It is **RECOMMENDED** that servers unable to process a particular content encoding return a 415 (Unsupported Media Type) status code.

3.7.7 Push-Mode Agent Requirements

A Push-Mode Agent that wishes to receive compressed/encoded requests from the ZIS **MUST** register its preference with the ZIS in the SIF_Register/SIF_Protocol property entitled Accept-Encoding in SIF_Name, providing an Accept-Encoding value in SIF_Value per HTTP 1.1 (the recommended compression algorithm for SIF is gzip). The agent **MUST** be prepared to handle an error SIF_Ack

from the ZIS when registering `Accept-Encoding` (`SIF_Error/SIF_Category` of 5 [Registration], `SIF_Error/SIF_Code` value of 10) if the ZIS cannot support at least one specified encoding and **SHOULD** re-attempt registration without `Accept-Encoding`.

Upon successful registration of an `Accept-Encoding` value, the agent **SHOULD** expect to receive requests from the ZIS encoded accordingly, but it **MAY** receive `identity`-encoded (unencoded) requests unless `identity` was explicitly excluded in the registered `Accept-Encoding` value.

3.7.8 Zone Integration Server Transport Requirements

A Zone Integration Server that receives a `SIF_Register/SIF_Protocol/SIF_Property` named `Accept-Encoding` in `SIF_Name` must fail the attempt to register if the ZIS does not support at least one of the specified encodings (`SIF_Error/SIF_Category` of 5 [Registration], `SIF_Error/SIF_Code` value of 10). While this property is typically registered by Push-Mode Agents, Pull-Mode Agents may also specify this property when registering. A ZIS **SHOULD** compress requests when contacting a Push-Mode Agent if the agent has previously registered that preference, but it **MAY** send uncompressed requests if the Push-Mode Agent did not explicitly exclude the `identity` encoding in its registered `Accept-Encoding` value.

Zone Integration Servers that support handling of compressed/encoded requests **SHOULD** return an `Accept-Encoding` header `SIF_Value` in the `SIF_ZoneStatus/SIF_SupportedProtocols/SIF_Protocol/SIF_Property` named `Accept-Encoding` in `SIF_Name`.

4 Messaging

This section documents the messaging and message handling protocols defined in SIF. A messaging protocol consists of sending a `SIF_Message` to initiate an operation, receiving back a `SIF_Ack`; a message handling protocol consists of processing an incoming `SIF_Message` and responding with a `SIF_Ack` and possibly sending follow-up `SIF_Message`s. This section is independent of transport layer details, aside from encryption and authentication level impacts associated with individual messages. Unless otherwise noted, all protocols assume successful communication over the appropriate transport layer; agent and ZIS implementations should also be prepared to handle transport layer errors and exceptions, directly or wrapped in a `SIF_Ack/SIF_Error` by underlying code.

Note: The diagrams in this section are provided to clarify the steps in each protocol. If there is a discrepancy between a table and a diagram, the table is to take precedence in all cases.

4.1 Agent Protocols

4.1.1 Agent Messaging Protocols

This section documents how Agents should send individual messages, and the resulting post-conditions upon success or failure, along with any necessary steps to take. These correspond to each of the operations an Agent can initiate.

4.1.1.1 SIF_Register

An Agent must register with the ZIS to participate in a Zone. To do so, it sends a `SIF_Register` message. An Agent may at any time re-register by sending another `SIF_Register` message. The ZIS updates the Agent's registered settings accordingly.

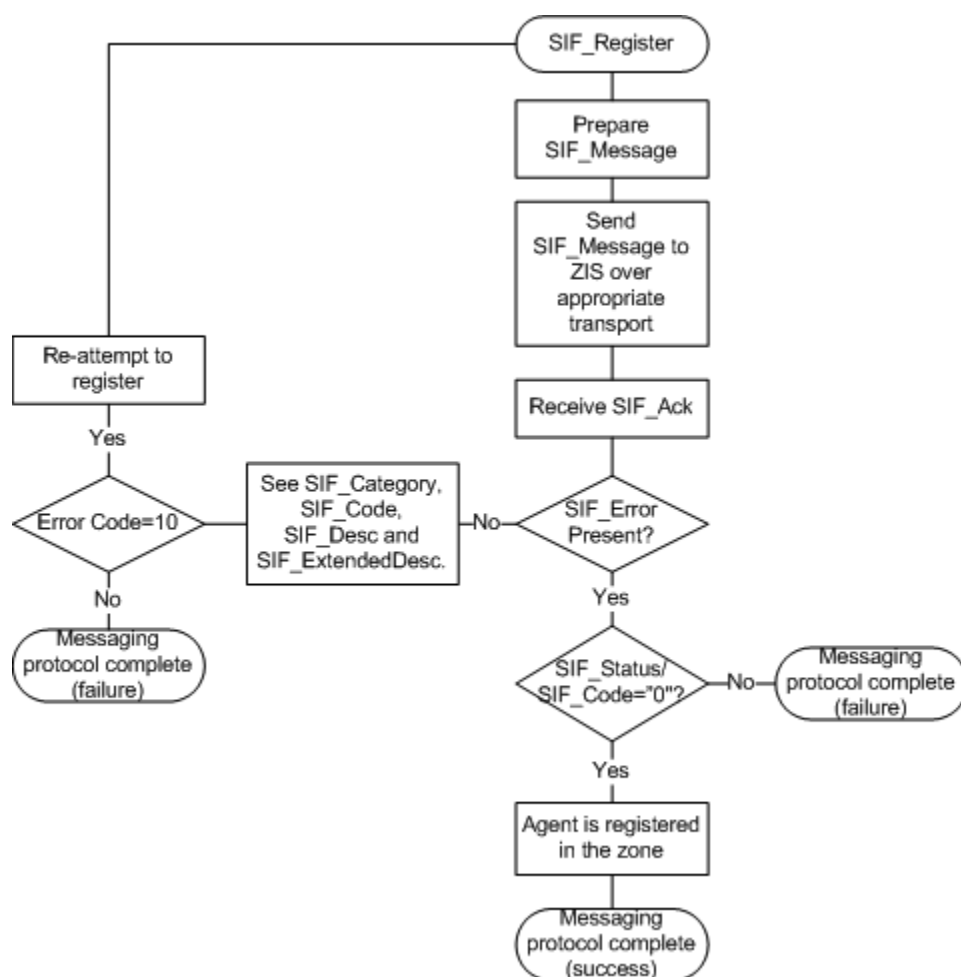


Figure 4.1.1.1-1: SIF_Register Agent Message Protocol

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_Register message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place your Agent's name, supported versions and maximum buffer size for receiving messages into SIF_Name , SIF_Version and SIF_MaxBufferSize , respectively. Specify whether your Agent is Push- or Pull-mode in SIF_Mode . If SIF_Mode is Push, specify the protocol information for the ZIS to use when delivering messages to your agent in SIF_Protocol ; optional compression settings may be included in the Accept-Encoding SIF_Protocol/SIF_Property . If desired, supply optional information regarding your Agent and/or application in SIF_NodeVendor , SIF_NodeVersion , SIF_Application and SIF_Icon .	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	Your Agent is now registered in the Zone. The Access Control settings for your agent (SIF_AgentACL are in SIF_Status/SIF_Data).	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included. Note particularly category 5. If an Accept-Encoding SIF_Protocol/SIF_Property was specified, the ZIS may return error code 10 (ZIS does not support the requested Accept-Encoding value). Your agent SHOULD re-attempt registration without, or with another, Accept-Encoding value.	Messaging protocol complete (failure).

Table 4.1.1.1-1: *SIF_Register Protocol*

4.1.1.2 SIF_Unregister

An Agent removes itself from a Zone by sending a [SIF_Unregister](#) message to the ZIS. Successful completion of this operation removes all settings associated with the Agent, including the objects it is currently providing and subscribed to in the zone; the Agent's message queue is also deleted. Note that a successful [SIF_Unregister](#) message may, depending on the ZIS implementation, remove access control settings that have been manually configured by a Zone administrator and that may need to be re-configured for a subsequent successful [SIF_Register](#).

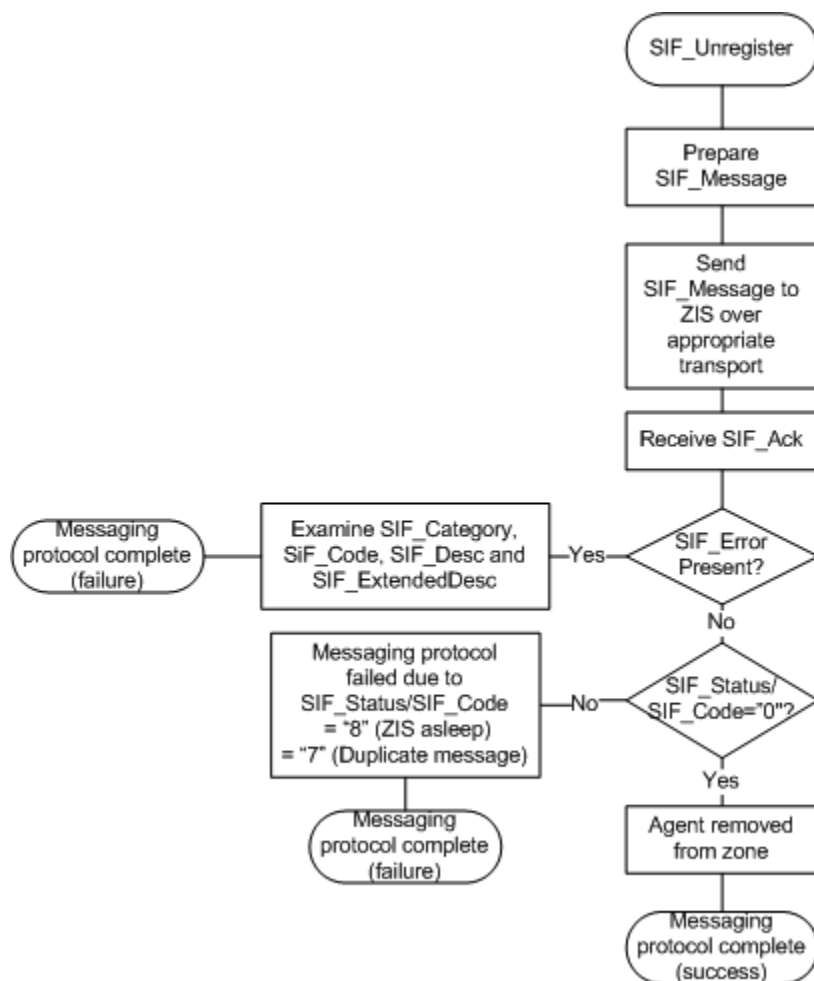


Figure 4.1.1.2-1: SIF_Unregister Agent Message Protocol

Step	Process	Flow Control
1	Prepare a <code>SIF_Message/SIF_Unregister</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply.	Send <code>SIF_Message</code> to ZIS over appropriate transport.
2	Receive <code>SIF_Ack</code> in response. Is <code>SIF_Error</code> present?	If yes, go to Step 6.
3	Is <code>SIF_Status/SIF_Code</code> 0?	If no, go to Step 5.
4	Your Agent is now removed from the Zone.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included.	Messaging protocol complete (failure).

Table 4.1.1.2-1: SIF_Unregister Protocol

4.1.1.3 SIF_Provide

An Agent registers with the ZIS to be the default Responder, or Provider, for one or more SIF objects in one or more contexts by sending a

`SIF_Provide` message to the ZIS. The Agent must have access control rights at the ZIS to successfully register as a Provider for an object.

Note that upon successful completion of `SIF_Provide` that your Agent is still the Provider of any objects for which it was previously registered as the Provider. To unregister as the Provider of given objects, use `SIF_Unprovide`. To replace all objects your Agent provides in one operation, use `SIF_Provision`.

As of version 2.0 of this specification, `SIF_Provision` is the preferred method for registering an Agent as a Provider, and provisioning an Agent in general. Support for `SIF_Provide` may be removed in a future major release of this specification.

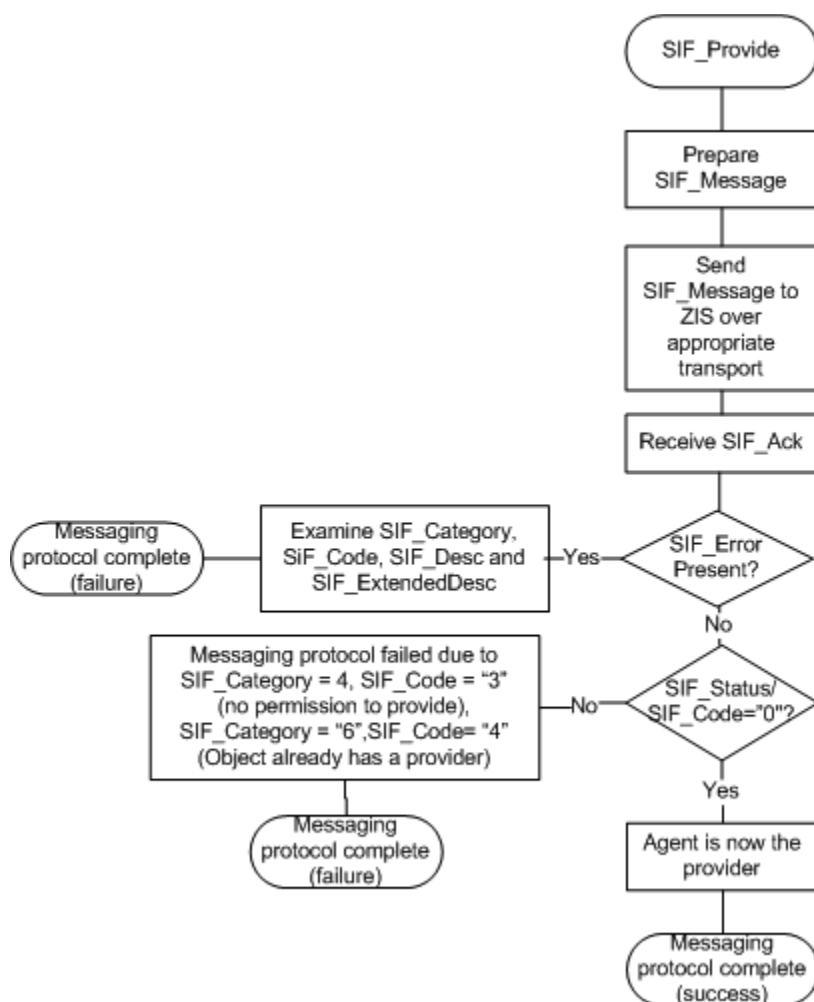


Figure 4.1.1.3-1: `SIF_Provide` Agent Message Protocol

Step	Process	Flow Control
1	Prepare a <code>SIF_Message/SIF_Provide</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. For each object your Agent would like to provide, place a <code>SIF_Object</code> element with an <code>ObjectName</code> and optionally one or more <code>SIF_Context</code> names (which default to <code>SIF_Default</code> if omitted). Your Agent's support for <code>SIF_ExtendedQuery</code> can be specified in <code>SIF_ExtendedQuerySupport</code> for each object.	Send <code>SIF_Message</code> to ZIS over appropriate transport.
2	Receive <code>SIF_Ack</code> in response. Is <code>SIF_Error</code> present?	If yes, go to Step 6.
3	Is <code>SIF_Status/SIF_Code</code> 0?	If no, go to Step 5.
4	Your Agent is now the Provider of each of the objects specified in the <code>SIF_Provide</code> message, in the applicable context(s). Any request sent by an Agent for one of these objects without explicitly specifying a particular Responder in <code>SIF_Header/SIF_DestinationId</code> will be placed in your Agent's message queue.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included. Note particularly category 4, code 3 (no permission to provide) and category 6, code 4 (object already has a Provider).	Messaging protocol complete (failure).

Table 4.1.1.3-1: `SIF_Provide` Protocol

4.1.1.4 SIF_Unprovide

Your Agent unregisters with the ZIS as the default Responder, or Provider, for one or more SIF objects in one or more contexts by sending a

`SIF_Unprovide` message to the ZIS. Note that any `SIF_Request`s for these objects already pending in your Agent's queue will still be delivered.

As of version 2.0 of this specification, `SIF_Provision` is the preferred method for unregistering an Agent as a Provider, and provisioning an Agent in general. Support for `SIF_Unprovide` may be removed in a future major release of this specification.

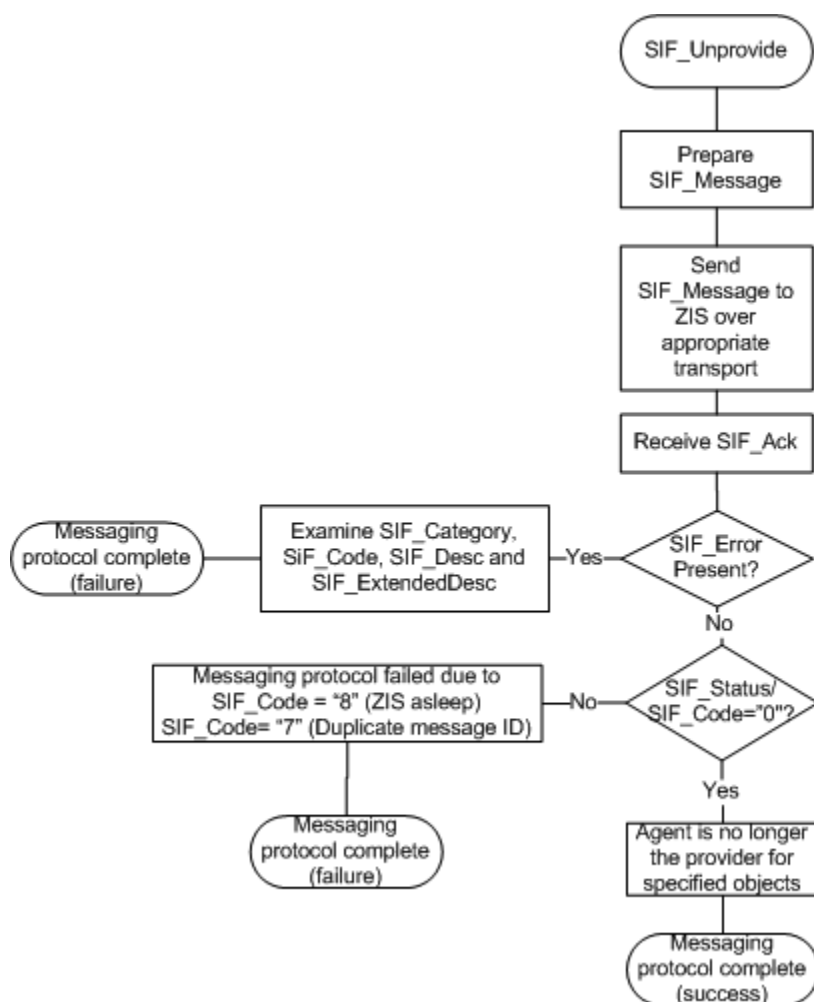


Figure 4.1.1.4-1: `SIF_Unprovide` Agent Message Protocol

Step	Process	Flow Control
1	Prepare a <code>SIF_Message/SIF_Unprovide</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. For each object your Agent would like to provide, include a <code>SIF_Object</code> element with an <code>ObjectName</code> and optionally one or more <code>SIF_Context</code> names (which default to <code>SIF_Default</code> if omitted).	Send <code>SIF_Message</code> to ZIS over appropriate transport.
2	Receive <code>SIF_Ack</code> in response. Is <code>SIF_Error</code> present?	If yes, go to Step 6.
3	Is <code>SIF_Status/SIF_Code</code> 0?	If no, go to Step 5.
4	Your Agent is no longer the Provider of each of the objects specified in the <code>SIF_Unprovide</code> message, in the applicable context(s). <code>SIF_Requests</code> will no longer be routed to your Agent by default, but this does not prevent other Agents from sending requests directly to your Agent (if permitted by access control rights).	Messaging protocol complete (success).
5	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).

Step	Process	Flow Control
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.4-1: *SIF_Unprovide Protocol*

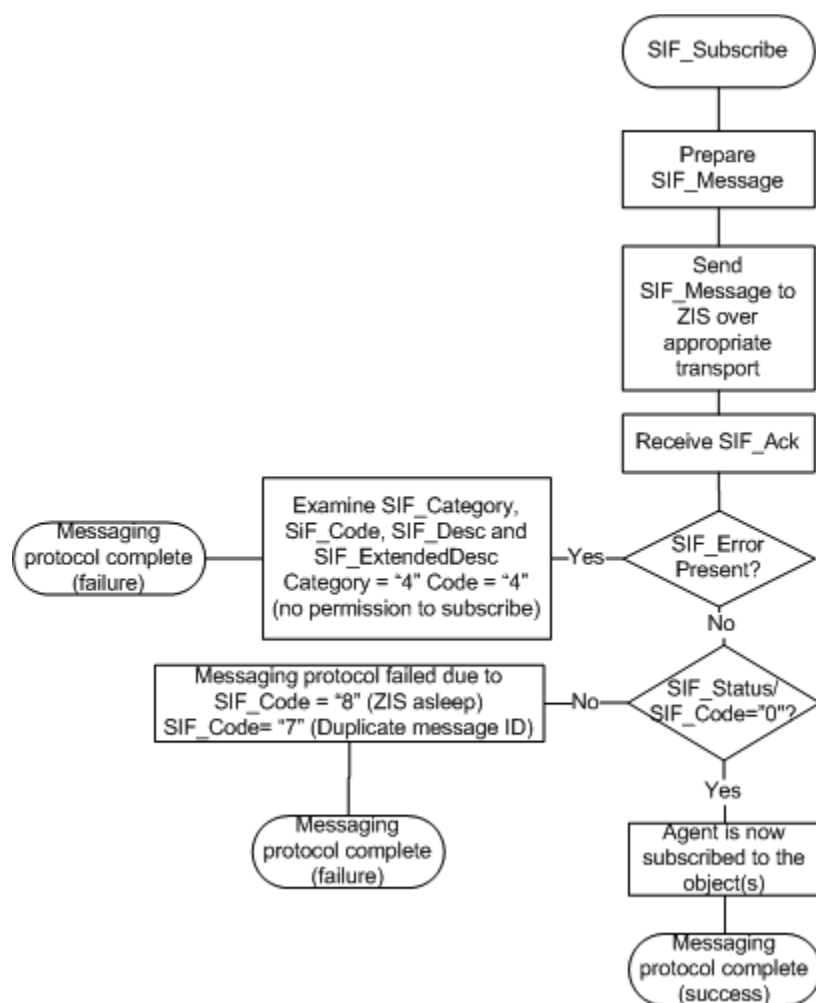
4.1.1.5 SIF_Subscribe

An Agent registers with the ZIS to receive [SIF_Events](#) for one or more SIF objects in one or more contexts by sending a

[SIF_Subscribe](#) message to the ZIS. The Agent must have access control rights at the ZIS to successfully subscribe to events for an object.

Note that upon successful completion of [SIF_Subscribe](#) that your Agent is still subscribed to objects to which it had previously subscribed. To unregister as a Subscriber of given objects, use [SIF_Unsubscribe](#). To replace all objects to which your Agent subscribes in one operation, use [SIF_Provision](#).

As of version 2.0 of this specification, [SIF_Provision](#) is the preferred method for registering an Agent as a Subscriber, and provisioning an Agent in general. Support for [SIF_Subscribe](#) may be removed in a future major release of this specification.

Figure 4.1.1.5-1: *SIF_Subscribe Agent Message Protocol*

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_Subscribe message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. For each object your Agent would like to subscribe to, place a SIF_Object element with an ObjectName and optionally one or more SIF_Context names (which default to SIF_Default if omitted).	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.

Step	Process	Flow Control
3	Is <code>SIF_Status/SIF_Code</code> 0?	If no, go to Step 5.
4	Your Agent is now subscribed to each of the objects specified in the <code>SIF_Subscribe</code> message, in the specified context(s) if included. Any <code>SIF_Events</code> for these objects will be placed in your Agent's queue.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included. Note particularly category 4, code 4 (no permission to subscribe).	Messaging protocol complete (failure).

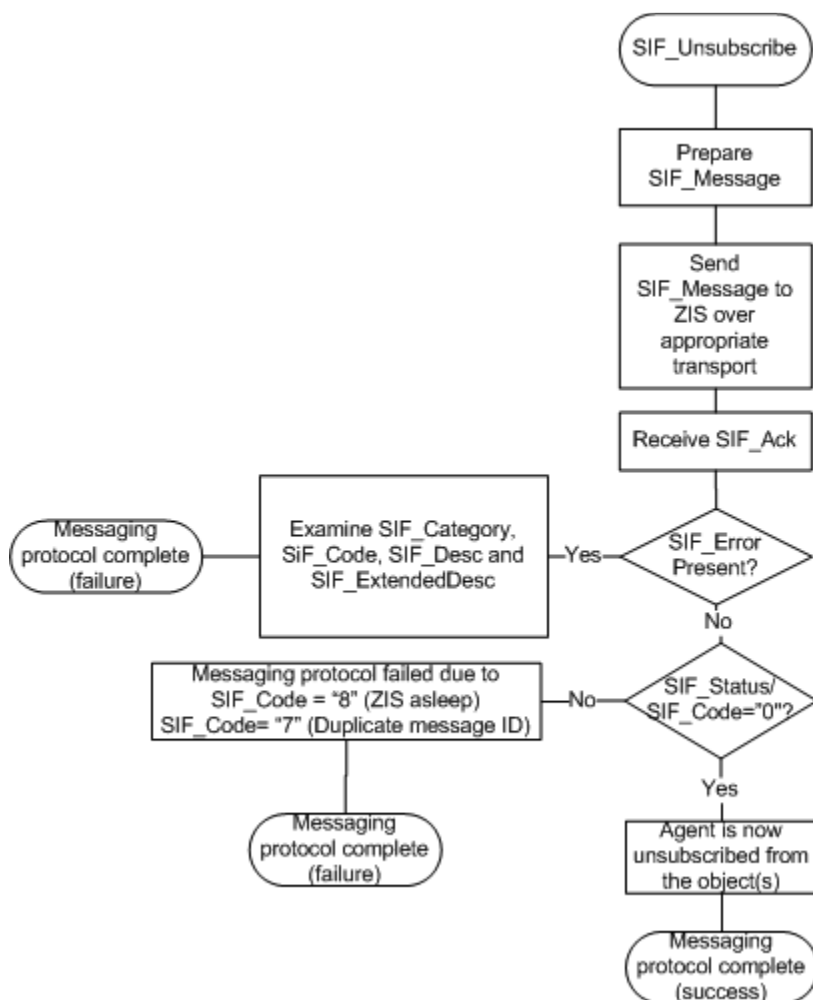
Table 4.1.1.5-1: *SIF_Subscribe Protocol*

4.1.1.6 SIF_Unsubscribe

To stop receiving `SIF_Events` for one or more objects in one or more contexts, an Agent sends a

`SIF_Unsubscribe` message to the ZIS. Note that if there are events already pending in your Agent's queue for these objects, they will still be delivered after a successful `SIF_Unsubscribe`.

As of version 2.0 of this specification, `SIF_Provision` is the preferred method for unregistering an Agent as a Subscriber, and provisioning an Agent in general. Support for `SIF_Unsubscribe` may be removed in a future major release of this specification.

Figure 4.1.1.6-1: *SIF_Unsubscribe Agent Message Protocol*

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_Unsubscribe message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. For each object your Agent would like to cease receiving events, include a SIF_Object element with an ObjectName and optionally one or more SIF_Context names (which default to SIF_Default if omitted).	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	Your Agent is now unsubscribed from each of the objects specified in the SIF_Unsubscribe message, in the applicable context(s). SIF_Events for these objects will cease to be placed in your Agent's queue.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.6-1: *SIF_Unsubscribe Protocol*

4.1.1.7 SIF_Provision

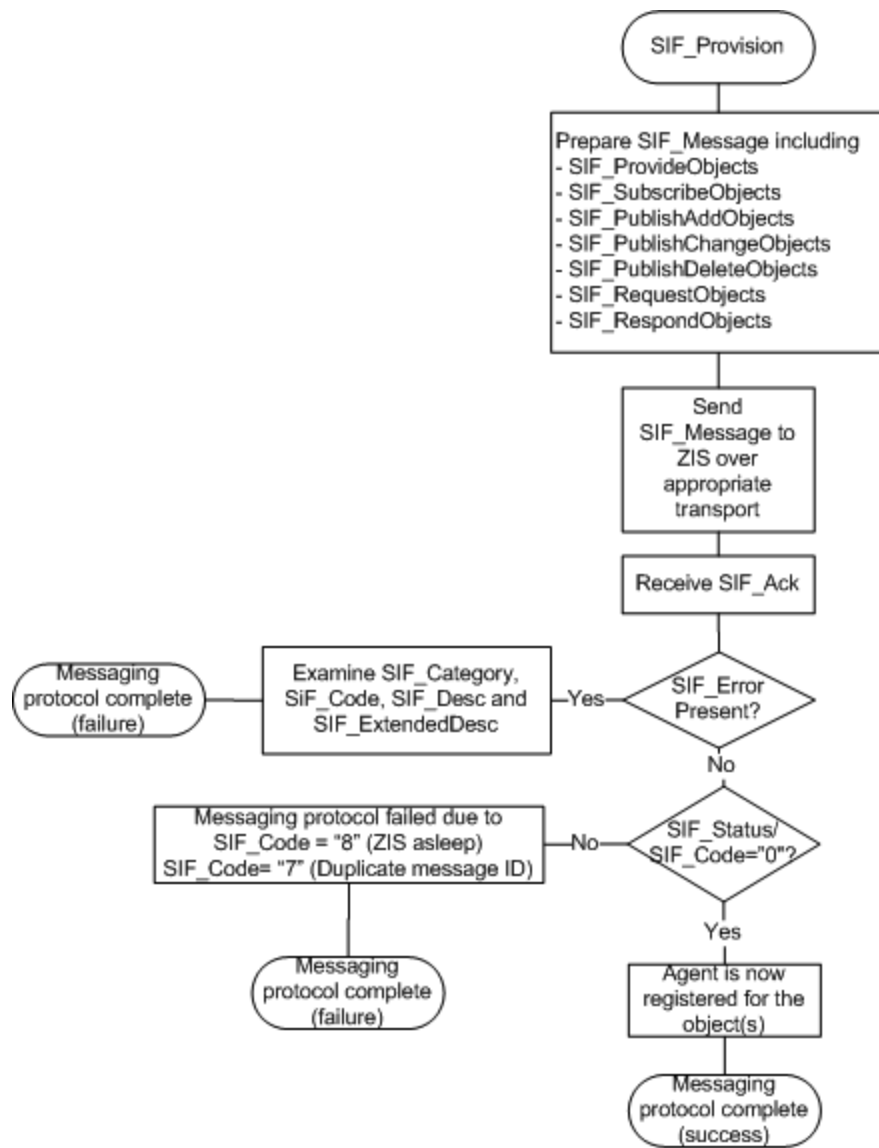
The [SIF_Provision](#) message provides an Agent the ability to register the objects it provides and to which it subscribes in a single operation, replacing whatever settings the ZIS has on record for the Agent. In effect it is an alternative to [SIF_Provide](#), [SIF_Unprovide](#), [SIF_Subscribe](#) and [SIF_Unsubscribe](#), though an Agent may choose to use this message or those, or a combination of these messages.

This message also allows the Agent to fully describe the operations it will perform in a Zone beyond those that can be communicated with [SIF_Provide](#) and [SIF_Subscribe](#), including the types of events it will publish, the requests to which it will respond with or without being the Provider for requested objects, and the objects for which it sends requests.

The Agent must have the appropriate access control settings to successfully register any of the corresponding information included in [SIF_Provision](#). Note that the list of access control settings can be determined by examining the [SIF_AgentACL](#) object returned in the [SIF_Register](#) or [SIF_GetAgentACL](#) message protocols.

Note that [SIF_Provision](#) will also fail if the Agent is attempting to provide an object that is already provided by another Agent in the applicable Zone Context. The list of Providers in a Zone can be found in [SIF_ZoneStatus](#).

As of version 2.0 of this specification, [SIF_Provision](#) is the preferred method for provisioning an Agent. Support for [SIF_Subscribe](#), [SIF_Unsubscribe](#), [SIF_Provide](#) and [SIF_Unprovide](#) may be removed in a future major release of this specification.

*Figure 4.1.1.7-1: SIF_Provision Agent Message Protocol*

Step	Process	Flow Control
1	<p>Prepare a SIF_Message/SIF_Provision message with SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp; other SIF_Header elements do not apply.</p> <p>Include SIF_ProvideObjects and for each object your Agent would like to provide, place a SIF_Object element with an ObjectName and optionally one or more SIF_Context names (which default to SIF_Default if omitted). Your Agent can also state its support for SIF_ExtendedQuery in SIF_ExtendedQuerySupport, which defaults to false.</p> <p>Include SIF_SubscribeObjects and for each object to which your Agent would like to subscribe, place a SIF_Object element with an ObjectName and optionally one or more SIF_Context names (which default to SIF_Default if omitted).</p> <p>Include SIF_PublishAddObjects, SIF_PublishChangeObjects and SIF_PublishDeleteObjects elements and include a SIF_Object element with an ObjectName in the respective sections for each event type your agent publishes with regard to that object. Optionally specify for each object one or more SIF_Context names (which default to SIF_Default if omitted).</p> <p>Include SIF_RequestObjects and for each object your Agent requests, place a SIF_Object element with an ObjectName and optionally one or more SIF_Context names (which default to SIF_Default if omitted). Your Agent can also state its support for SIF_ExtendedQuery in SIF_ExtendedQuerySupport, which defaults to false.</p> <p>Include SIF_RespondObjects and for each object for which your Agent processes requests (including those listed in SIF_ProvideObjects), include a SIF_Object element with an ObjectName and optionally one or more SIF_Context names (which default to SIF_Default if omitted). Your Agent can also state its support for SIF_ExtendedQuery in SIF_ExtendedQuerySupport, which defaults to false.</p> <p>Optionally, if your agent provides SIF Zone Services, include SIF_ProvideService and for each service your agent would like to provide, place a SIF_Service element with a ServiceName and optionally one or more SIF_Context names (which default to SIF_Default if omitted).</p> <p>Optionally, if your agent responds to SIF_ServiceInput requests, include SIF_RespondService for each service for which your agent processes requests (including those listed in SIF_ProvideService), include a SIF_Service element with a ServiceName and optionally one or more SIF_Context names (which default to SIF_Default if omitted).</p> <p>Optionally, if your agent supports sending SIF_ServiceInput requests, include SIF_RequestService and for each service your agent requests, place a SIF_Service element with a ServiceName and optionally one or more SIF_Context names (which default to SIF_Default if omitted). Your agent can also state its support for SIF_ExtendedQuery in SIF_ExtendedQuerySupport, which defaults to false.</p> <p>Optionally, if your agent supports subscribing to SIF Zone Service notifications, include SIF_SubscribeService and for each object to which your agent would like to subscribe, place a SIF_Service element with a ServiceName and optionally one or more SIF_Context names (which default to SIF_Default if omitted).</p>	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	Your Agent is now registered with the corresponding settings in the Zone. Any previously recorded settings with regard to the operations your Agent performs have been replaced.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.7-1: SIF_Provision Protocol

4.1.1.8 SIF_Event

When an application adds, changes or deletes data represented in one or more Zone Contexts, its Agent **SHOULD** publish the corresponding Add, Change or Delete SIF_Event to the Zone. Upon successful delivery of a SIF_Event to the ZIS, the ZIS places the event in the queue for any Agents subscribed to events for the object, including your Agent if it is a subscriber.

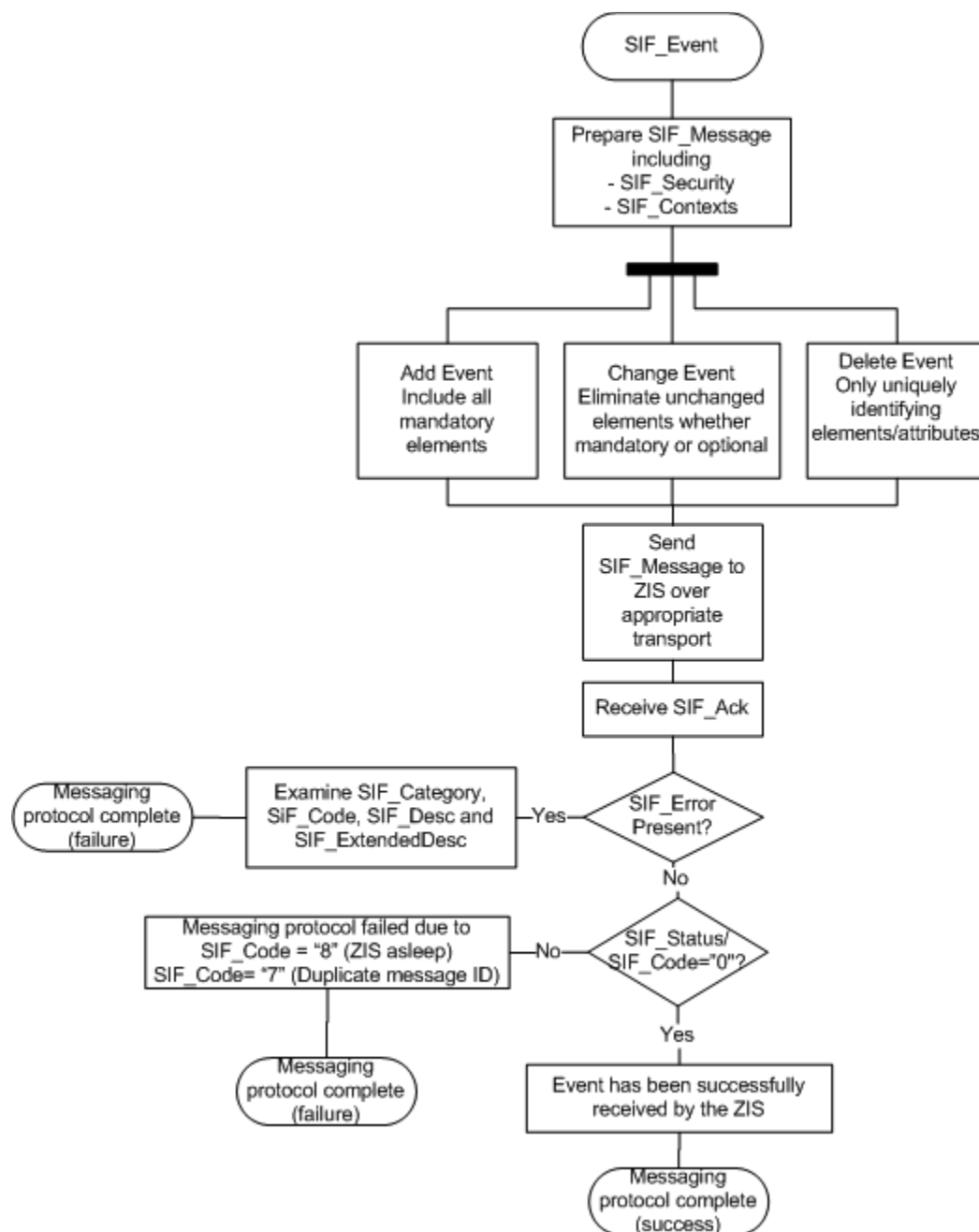


Figure 4.1.1.8-1: SIF_Event Agent Message Protocol

Step	Process	Flow Control
1	<p>Prepare a SIF_Message/SIF_Event message with SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp. If your Agent would like to indicate minimum encryption and/or authentication requirements for Agents receiving this SIF_Event; supply SIF_Security with the appropriate settings; use an equally secure channel when communicating with the ZIS, if desired. If this event specifically applies to one or more contexts, place them in SIF_Contexts; if omitted, the context is SIF_Default.</p> <p>Specify the name of the object that is being added, changed or deleted in SIF_EventObject/@ObjectName. Place the type of event in SIF_EventObject/@Action and place the object in SIF_EventObject. For an Add event, this MUST be the complete object with all mandatory elements present. If the agent wishes to indicate that a particular optional element is supported but has no value, the element MAY be included as empty, with <code>xsi:nil</code> set to <code>true</code> if necessary.</p> <p>For a Change event, all unchanged elements, whether mandatory or optional SHOULD be omitted from the object. Optional elements that have been deleted MAY be included as empty, with <code>xsi:nil</code> set to <code>true</code> if necessary. For each list of repeatable elements in the object that has changed, include the whole list if the list type indicated is <code>List</code>. If the list type is <code>ActionList</code>, the agent MAY include only those elements in the list that have been added, changed or deleted. If an element has been deleted from an <code>ActionList</code>, the element MUST be included with at least its key attribute(s) and/or element(s) specified, and include a SIF_Action attribute value of <code>Delete</code> on the deleted child element in the list. Omitting an element in an <code>ActionList</code> indicates that it has been unchanged in the event. Refer to the Data Model section of the specification, Lists/Repeatable Elements, for more details on <code>ActionLists</code> and <code>Lists</code>.</p> <p>For a Delete event, only elements/attributes that identify the object sufficiently for deletion SHOULD be included. This set of identifying elements/attributes are typically communicated by the mandatory root attributes of an object, which MUST be included.</p>	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	The event has been successfully received by the ZIS. It will be placed in the queue of any Agents registered as subscribers to events for the given object.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.8-1: *SIF_Event Protocol*

4.1.1.9 SIF_Request

An Agent can request data from another Agent at any time by sending a [SIF_Request](#) message. Agents use one of two query mechanisms in requests. SIF's default query mechanism, [SIF_Query](#), is used to request objects of a given type, matching optional query conditions, optionally returning a subset of object elements. [SIF_ExtendedQuery](#) is used to select elements from one or more objects, joined together, if necessary, on `RefId`-based conditions. Before delivering a request with a [SIF_ExtendedQuery](#) to a Responder, the ZIS checks that the Responder supports [SIF_ExtendedQuery](#) for all referenced objects.

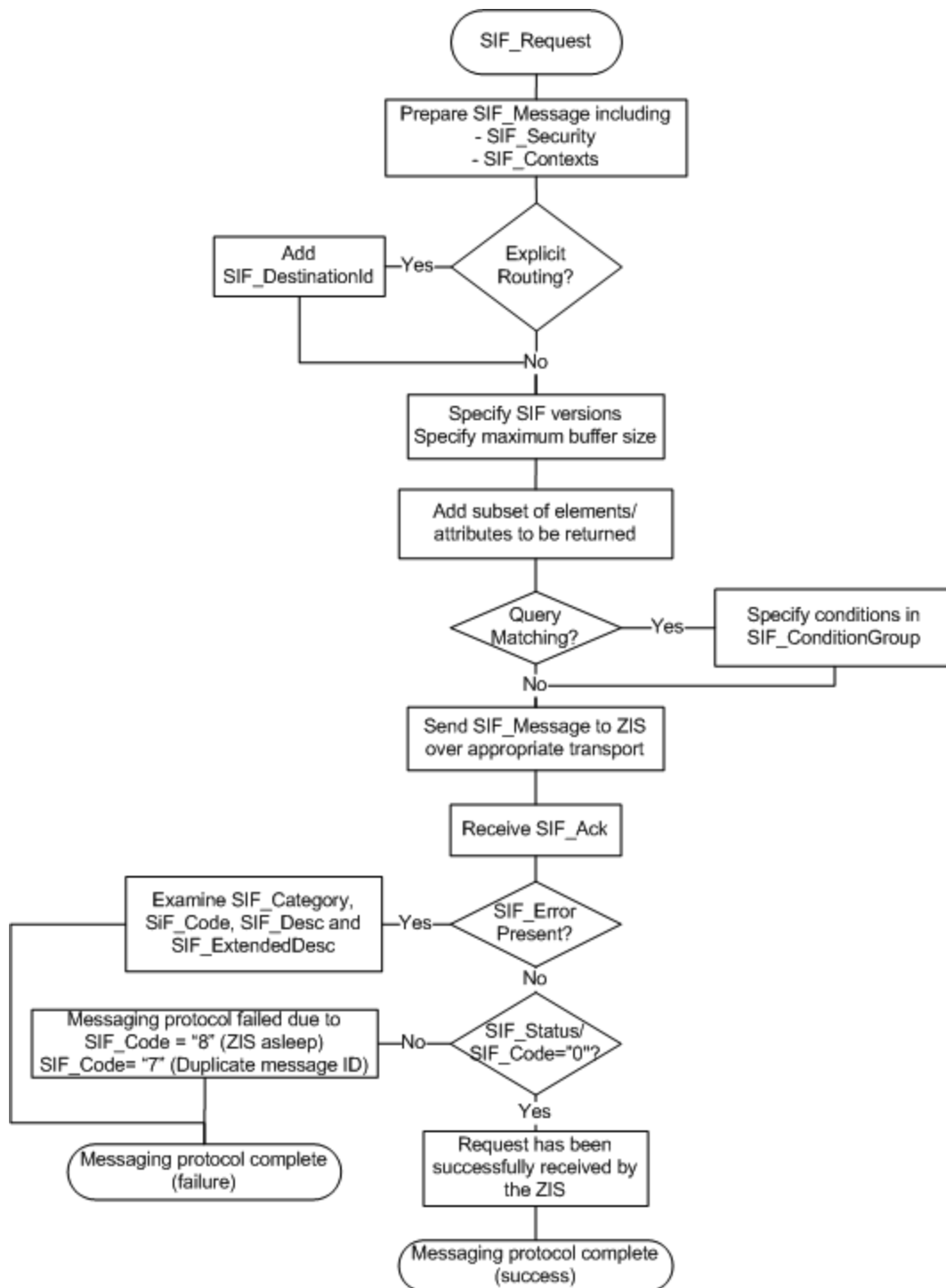


Figure 4.1.1.9-1: SIF_Request Agent Message Protocol

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_Request message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp . If your Agent would like to indicate minimum encryption and/or authentication requirements for Agents receiving this SIF_Request , supply SIF_Security with the appropriate settings; use an equally secure channel when communicating with the ZIS, if desired. If this request is associated with a context, specify a single SIF_Context in SIF_Contexts ; if omitted, the context is SIF_Default . If your Agent would like to explicitly route this request to a given Agent, specify the Agent's Id in SIF_DestinationId . Specify the SIF versions the responder may choose from when returning data in SIF_Version . Each version specified MUST be registered at the ZIS as supported by your Agent. Specify the maximum buffer size the Responder must respect when sending SIF_Response packets; this MUST be less than or equal to the SIF_MaxBufferSize with which your Agent registered with the ZIS.	If using SIF_ExtendedQuery , go to step 3; otherwise go to step 2.
2	In SIF_Query , specify the object name being requested in SIF_QueryObject/@ObjectName . Optionally specify the subset of elements/attributes to be returned from each object in SIF_QueryObject/SIF_Element ; note that parent elements of specified elements/attributes are returned as well. If your Agent would like to specify query matching conditions, include SIF_ConditionGroup . Alternately an example of an object allowed for use in query-by-example can be placed in SIF_Example .	Send SIF_Message to ZIS over appropriate transport. Go to step 4.
3	Include a SIF_ExtendedQuery . If your Agent did not specify SIF_DestinationId , the SIF_Request will be routed to the Provider for SIF_From/@ObjectName . If your Agent would like to override this routing mechanism, include SIF_DestinationProvider set to the object name for which the ZIS will determine the Provider and route the request accordingly.	Send SIF_Message to ZIS over appropriate transport.
4	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 8.
5	Is SIF_Status/SIF_Code 0?	If no, go to Step 7.
6	The request has been successfully received by the ZIS. It will be placed in the queue of the appropriate Responder as specified in SIF_Header/SIF_DestinationId or determined by SIF_ExtendedQuery/SIF_From/@ObjectName OR SIF_ExtendedQuery/SIF_DestinationProvider .	Messaging protocol complete (success).
7	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
8	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.9-1: *SIF_Request Protocol*

4.1.1.10 SIF_Ping

An agent can "ping" the ZIS or check that it's online and/or "awake" by sending a

[SIF_Ping](#) message to the ZIS. If the agent receives a successful acknowledgement, the ZIS is awake; the ZIS may also reply that it is asleep. As a ZIS may be offline completely, Agents should be prepared to handle transport errors directly or wrapped in a [SIF_Ack/SIF_Error](#) by underlying code.

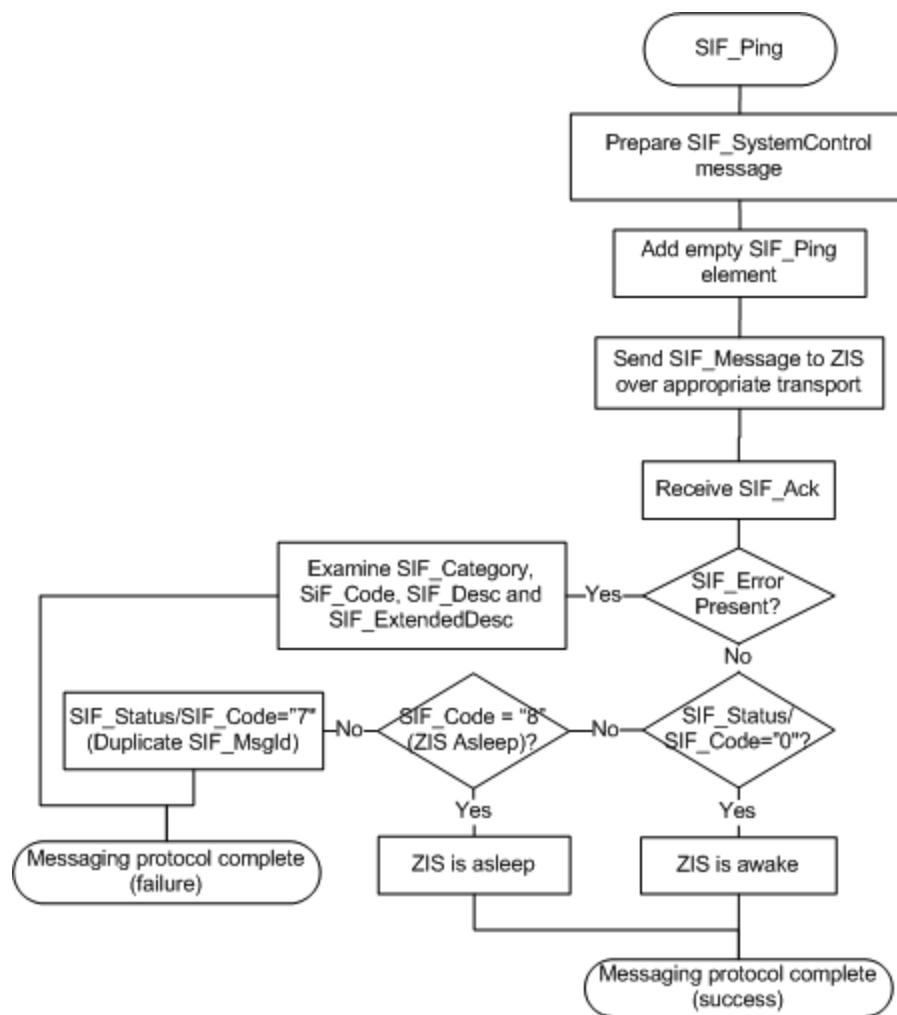


Figure 4.1.1.10-1: SIF_Ping Agent Message Protocol

Step	Process	Flow Control
1	Prepare a <code>SIF_SystemControl</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place an empty <code>SIF_Ping</code> element in <code>SIF_SystemControlData</code> .	Send <code>SIF_Message</code> to ZIS over appropriate transport.
2	Receive <code>SIF_Ack</code> in response. Is <code>SIF_Error</code> present?	If yes, go to Step 8.
3	Is <code>SIF_Status/SIF_Code</code> 0?	If no, go to Step 5.
4	The ZIS is awake.	Messaging protocol complete (success).
5	Is <code>SIF_Status/SIF_Code</code> 8 (ZIS is asleep)?	If no, go to Step 7.
6	The ZIS is asleep.	Messaging protocol complete (success).
7	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).
8	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included.	Messaging protocol complete (failure).

Table 4.1.1.10-1: SIF_Ping Protocol

4.1.1.11 SIF_Sleep

A Push-mode Agent can send a

`SIF_Sleep` message to the ZIS to change its state to "asleep," indicating that the ZIS should not send the Agent messages until it "wakes up" by sending a `SIF_Wakeup` message or re-registering with `SIF_Register`. A Pull-mode Agent can also change its state to "sleeping," but this has no effect other than indicating to other Agents via `SIF_ZoneStatus` that it is "sleeping" and not processing

messages in its queue. Sending a `SIF_Wakeup` or `SIF_GetMessage` will indicate that the Agent is "awake," as will re-registering with `SIF_Register`.

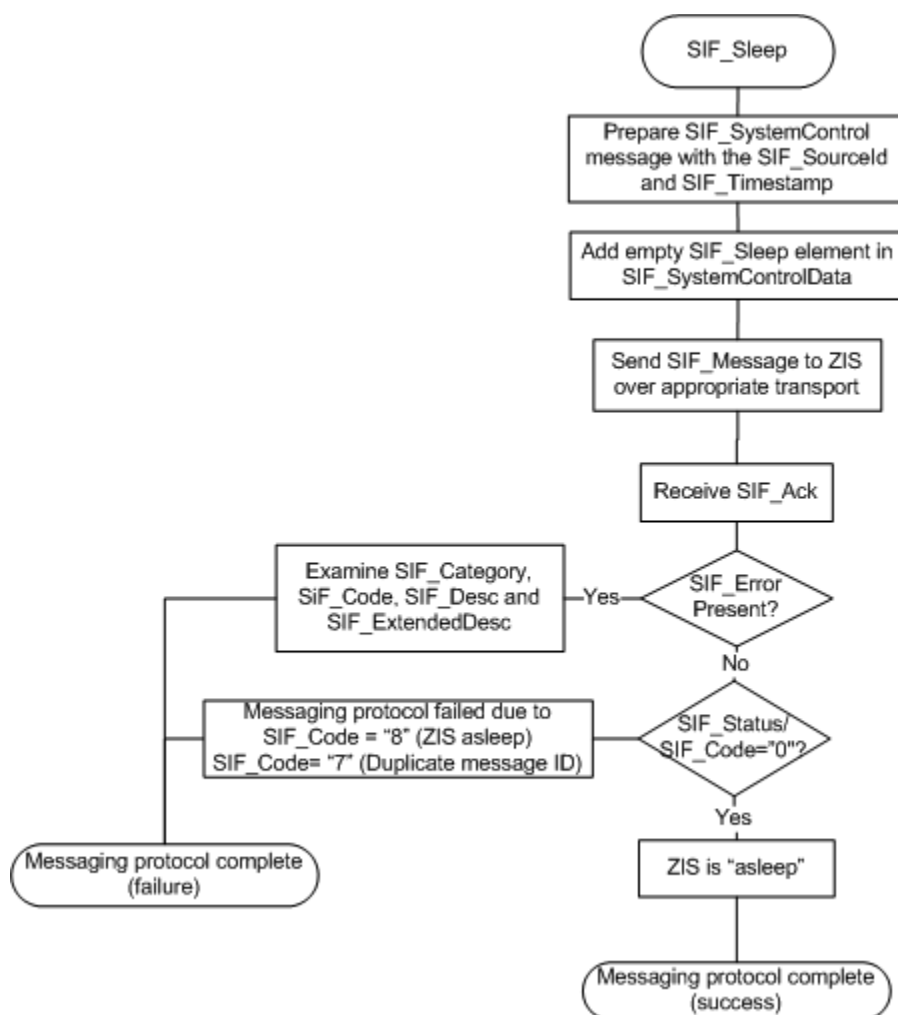


Figure 4.1.1.11-1: *SIF_Sleep Agent Message Protocol*

Step	Process	Flow Control
1	Prepare a <code>SIF_Message/SIF_SystemControl</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place an empty <code>SIF_Sleep</code> element in <code>SIF_SystemControlData</code> .	Send <code>SIF_Message</code> to ZIS over appropriate transport.
2	Receive <code>SIF_Ack</code> in response. Is <code>SIF_Error</code> present?	If yes, go to Step 6.
3	Is <code>SIF_Status/SIF_Code</code> 0?	If no, go to Step 5.
4	Your Agent's state has been set to "asleep" in the ZIS. This is reflected to other Agents in <code>SIF_ZoneStatus</code> and if your Agent is a Push-mode Agent, the ZIS will stop delivering messages to your Agent. To "wake up," send a <code>SIF_Wakeup</code> message, or re-register with <code>SIF_Register</code> . Pull-mode Agents may also send <code>SIF_GetMessage</code> .	Messaging protocol complete (success).
5	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included.	Messaging protocol complete (failure).

Table 4.1.1.11-1: *SIF_Sleep Protocol*

4.1.1.12 SIF_Wakeup

An Agent can send a

`SIF_Wakeup` message to the ZIS to change its state to "awake," whether sleeping or not; this state is available to other Agents via `SIF_ZoneStatus`. Upon success, the ZIS may begin delivering messages to a Push-mode Agent again, if previously sleeping.

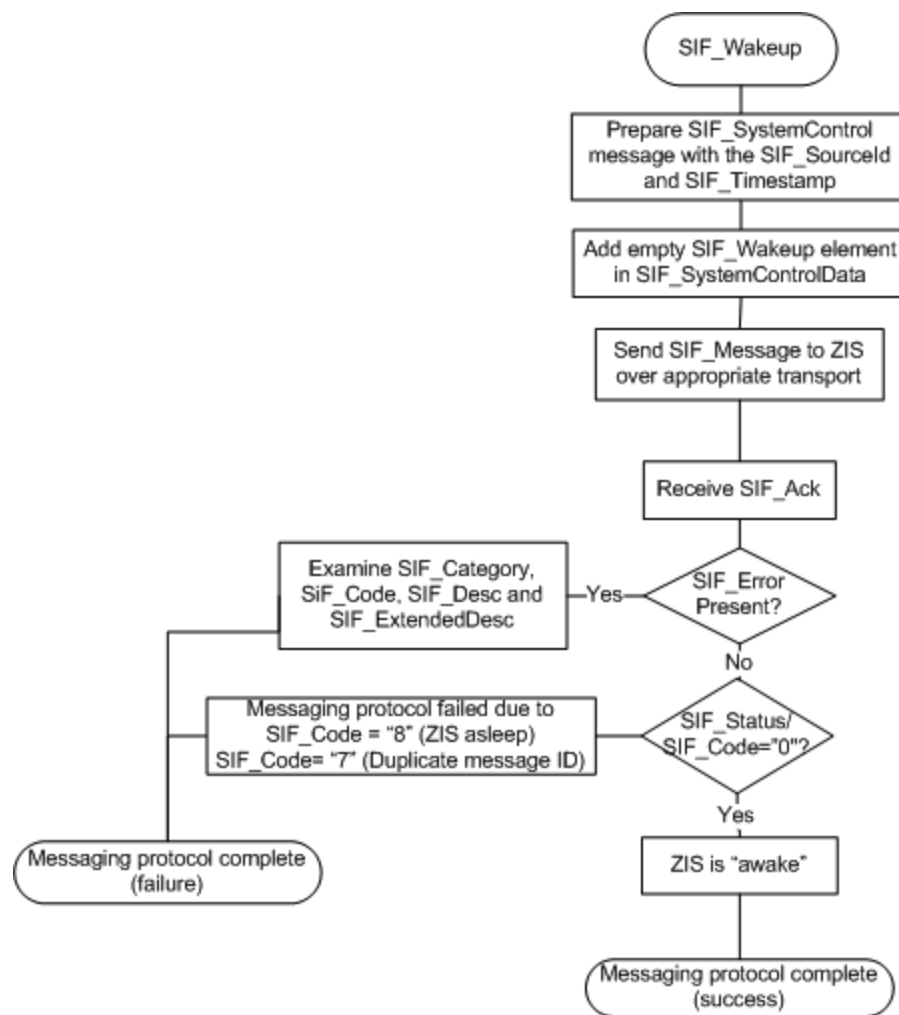


Figure 4.1.1.12-1: SIF_Wakeup Agent Message Protocol

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place an empty SIF_Wakeup element in SIF_SystemControlData .	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	Your Agent's state has been set to "awake" in the ZIS. This is reflected to other Agents in SIF_ZoneStatus and if your Agent is a Push-mode Agent and it was previously asleep, the ZIS will resume delivering messages to your Agent.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.12-1: SIF_Wakeup Protocol

4.1.1.13 SIF_GetZoneStatus

To retrieve the current status of the Zone (

[SIF_ZoneStatus](#)), send a [SIF_GetZoneStatus](#) message to the ZIS.

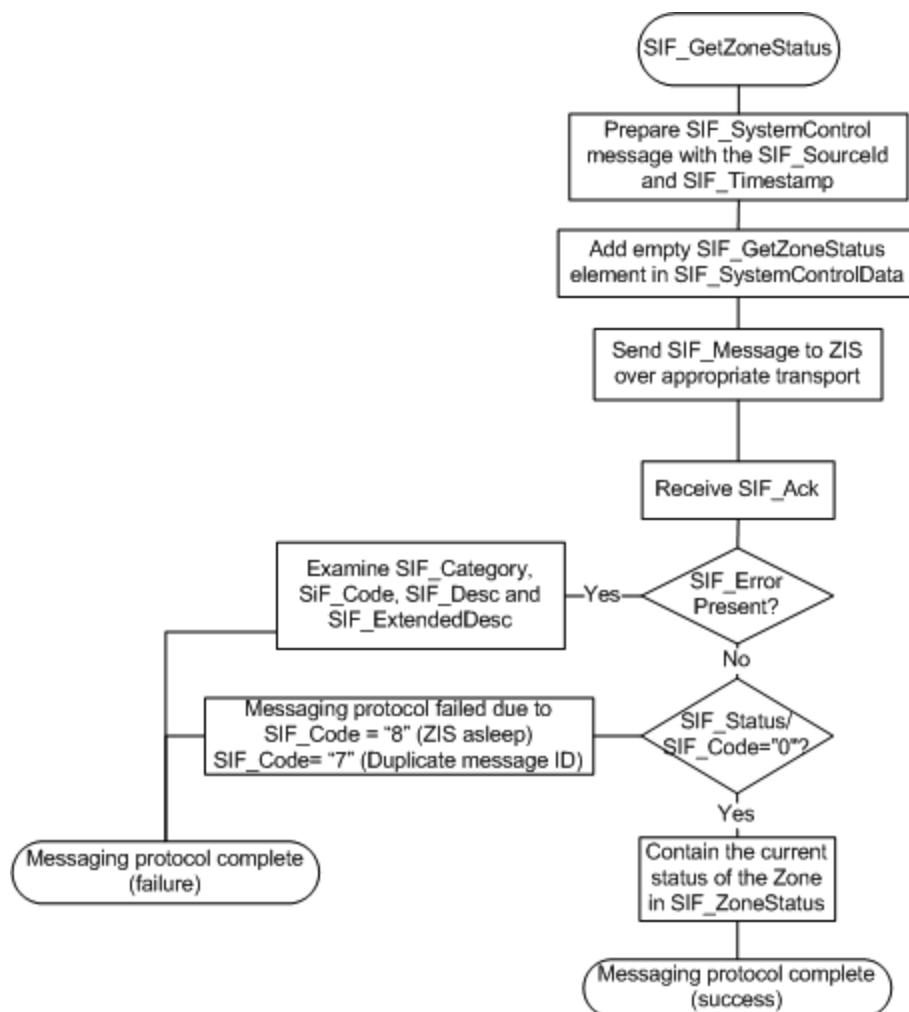


Figure 4.1.1.13-1: SIF_GetZoneStatus Agent Message Protocol

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place an empty SIF_GetZoneStatus element in SIF_SystemControlData .	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	SIF_Status/SIF_Data contains the current status of the Zone in SIF_ZoneStatus .	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.13-1: SIF_GetZoneStatus Protocol

4.1.1.14 SIF_GetAgentACL

To retrieve your Agent's current access control list settings from the ZIS (

[SIF_AgentACL](#)), send a [SIF_GetAgentACL](#) message to the ZIS.

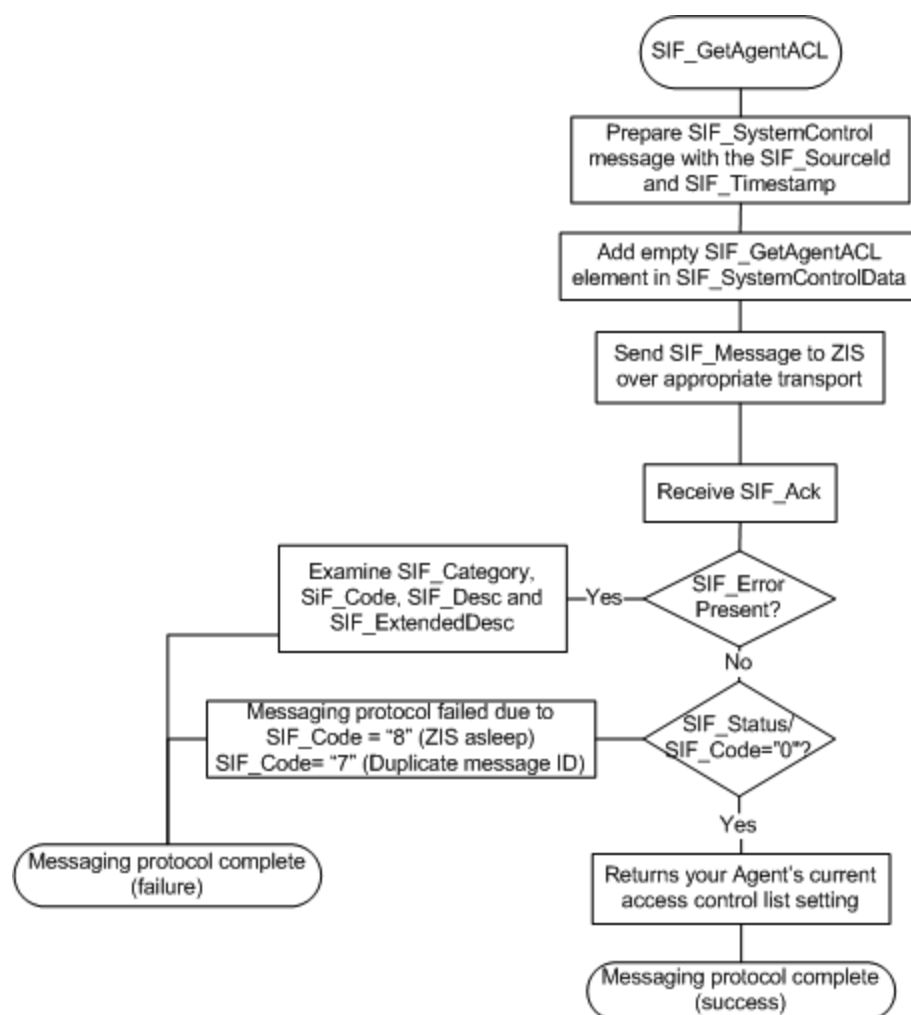


Figure 4.1.1.14-1: SIF_GetAgentACL Agent Message Protocol

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place an empty SIF_GetAgentACL element in SIF_SystemControlData .	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	SIF_Status/SIF_Data contains your Agent's current access control list settings in the Zone in SIF_AgentACL .	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.14-1: SIF_GetAgentACL Protocol

4.1.1.15 SIF_CancelRequests

Agents can request that a ZIS cancel [SIF_Requests](#), pending or in process, by sending a list of [SIF_RequestMsgIds](#) in a [SIF_CancelRequests](#) message. If an Agent abandons or restarts a data collection using [SIF_Requests](#), whether or not the response stream has started, it is [RECOMMENDED](#) that the Agent send one or more [SIF_CancelRequests](#) messages to the ZIS. Such data collections can place a heavy load on responding Agents, where often all data of a specific object type is requested, and cancelling requests may spare Zone resources. Cancelling of responses can also reduce the number of response packets the receiving/cancelling agent needs to process and discard.

If the cancelling Agent wishes to receive a "final" *SIF_Response* from the ZIS for each cancelled message, it can specify *Standard* in *SIF_CancelRequests/SIF_NotificationType*. If the cancelling Agent does not desire or require "final" *SIF_Responses*, the Agent can specify *None* in *SIF_NotificationType*.

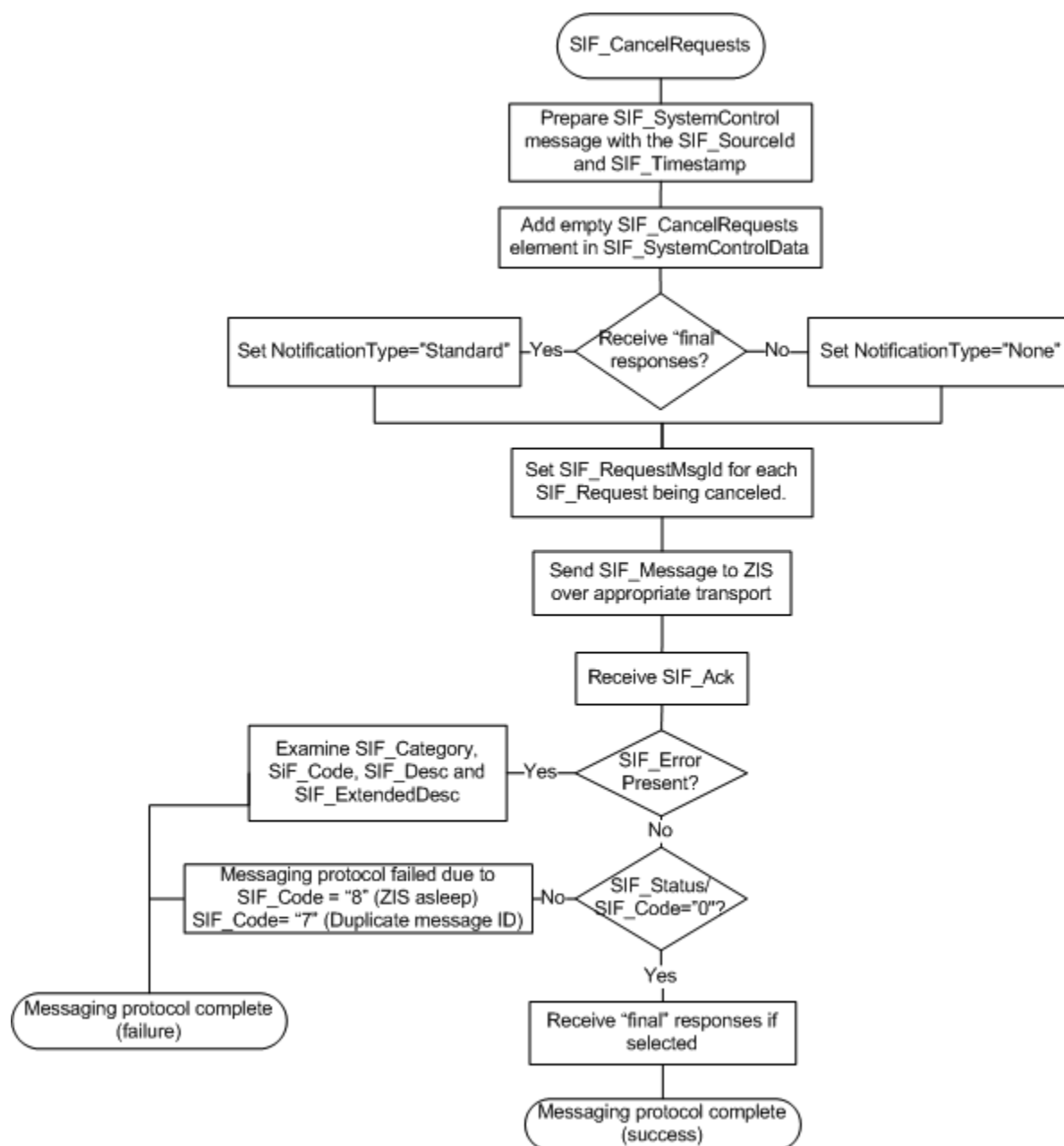


Figure 4.1.1.15-1: *SIF_CancelRequests* Agent Message Protocol

Step	Process	Flow Control
1	Prepare a <i>SIF_Message/SIF_SystemControl</i> message with <i>SIF_Header</i> containing a new GUID in <i>SIF_MsgId</i> , your Agent's Agent Id in <i>SIF_SourceId</i> and the current time in <i>SIF_Timestamp</i> ; other <i>SIF_Header</i> elements do not apply. Add a <i>SIF_CancelRequests</i> element in <i>SIF_SystemControlData</i> .	
2	Specify <i>Standard</i> in <i>NotificationType</i> if your Agent desires or requires a "final" <i>SIF_Response</i> be returned by the ZIS for each cancelled message (<i>SIF_Response/SIF_MorePackets</i> = No). Otherwise specify <i>None</i> .	
3	Add a <i>SIF_RequestMsgIds</i> element and add a child <i>SIF_RequestMsgId</i> element for each <i>SIF_Request</i> that the Agent wishes to cancel.	Send <i>SIF_Message</i> to ZIS over appropriate transport.
4	Receive <i>SIF_Ack</i> in response. Is <i>SIF_Error</i> present?	If yes, go to Step 8.
5	Is <i>SIF_Status/SIF_Code</i> 0?	If no, go to Step 7.

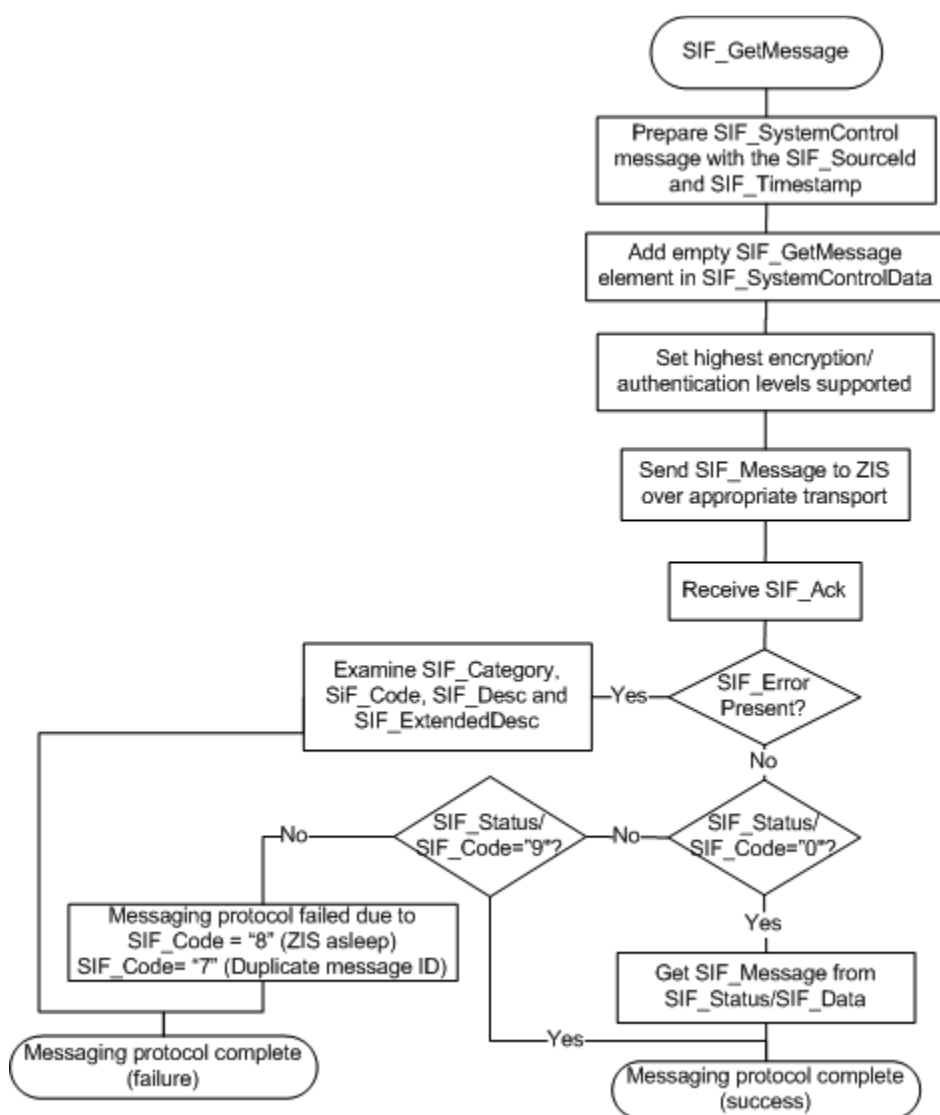
Step	Process	Flow Control
6	The ZIS has accepted the <code>SIF_CancelRequests</code> message. Your Agent will receive or not receive "final" <code>SIF_Responses</code> per the specified <code>NotificationType</code> .	Messaging protocol complete (success).
7	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).
8	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included.	Messaging protocol complete (failure).

Table 4.1.1.15-1: *SIF_CancelRequests Protocol*

4.1.1.16 SIF_GetMessage (Pull-Mode only)

Pull-mode Agents retrieve the next message in their queue by sending a

`SIF_GetMessage` message to the ZIS. Note that as individual messages may have specific minimum encryption/authentication levels attached to them by senders, a Pull-Mode Agent should always use the highest encryption/authentication levels it supports when contacting the ZIS to avoid individual messages being discarded when contacting the ZIS using lower encryption/authentication levels than might be required for receipt of a given message.

Figure 4.1.1.16-1: *SIF_GetMessage (Pull-Mode only) Agent Message Protocol*

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place an empty SIF_GetMessage element in SIF_SystemControlData .	Send SIF_Message to ZIS over appropriate transport. Always use the highest encryption/authentication levels that your Agent supports to maximize the number of messages that can be returned to your Agent.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 7.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	SIF_Status/SIF_Data contains the next SIF_Message in your agent's queue.	Messaging protocol complete (success). Process the returned SIF_Message according to Agent Message Handling Protocols below.
5	Is SIF_Status/SIF_Code 9?	If yes, there are no messages available for your Agent. Message processing complete (success).
6	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
7	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included. If a Push-mode Agent sends SIF_GetMessage , note particularly category 5, code 9 (agent registered in Push mode).	Messaging protocol complete (failure).

Table 4.1.1.16-1: [SIF_GetMessage](#) Protocol**4.1.1.17 [SIF_Ack](#) (Push-Mode)**

Push-Mode Agents end Selective Message Blocking (SMB) by sending a final [SIF_Ack](#) to the ZIS.

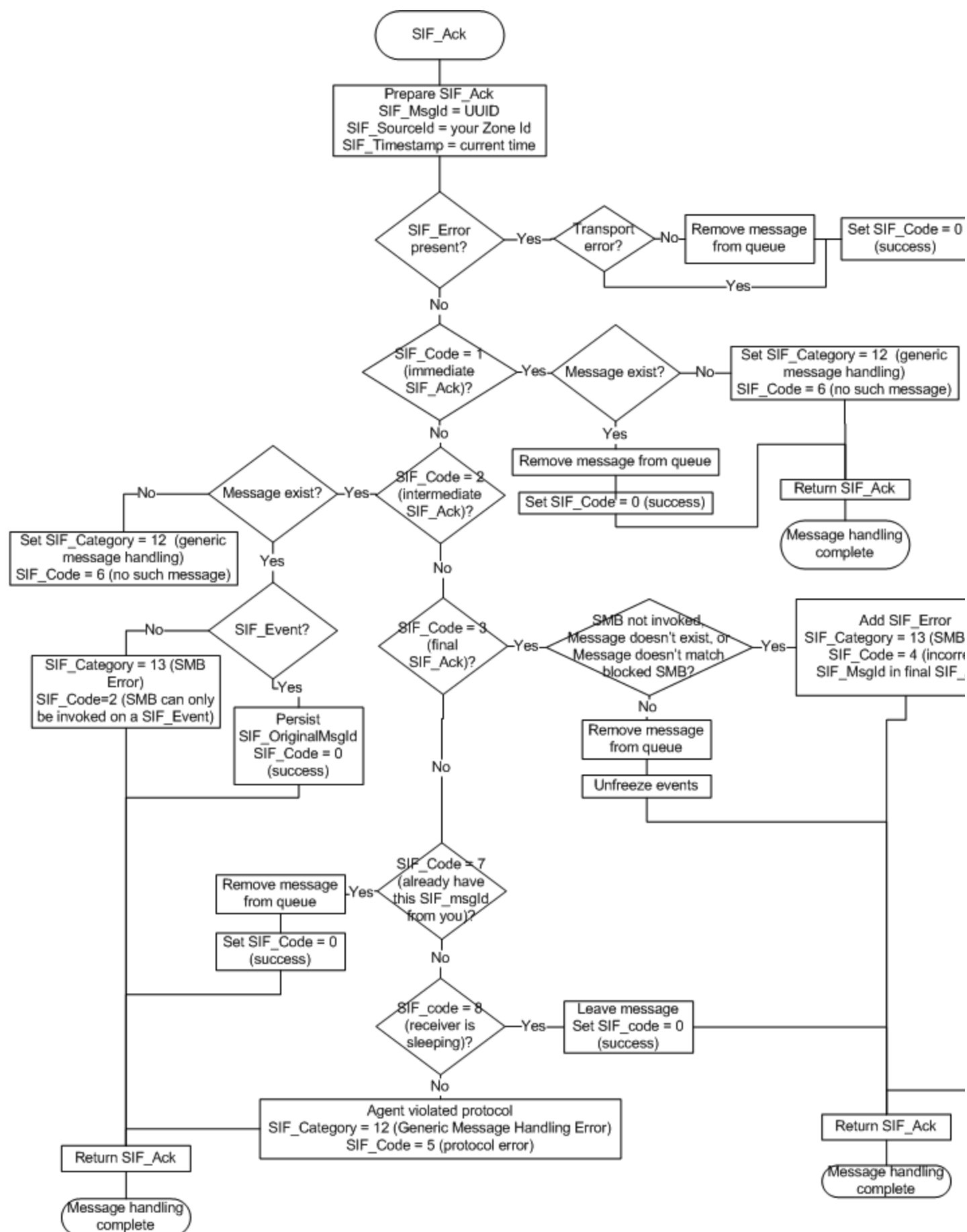


Figure 4.1.1.17-1: SIF_Ack (Push-Mode) Agent Message Protocol

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. From the message being unblocked/removed from the queue, place the SIF_Header/SIF_SourceId value into SIF_OriginalSourceId and place the SIF_Header/SIF_MsgId value into SIF_OriginalMsgId . Place 3 (final SIF_Ack) into SIF_Code/SIF_Data .	Send SIF_Message to ZIS over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 0?	If no, go to Step 5.
4	The referenced message has been unblocked and removed from your Agent's queue. The ZIS resumes delivery of events to your Agent.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.1.1.17-1: *SIF_Ack Protocol (Push-Mode)***4.1.1.18 SIF_Ack (Pull-Mode)**

Pull-mode Agents acknowledge messages received in response to [SIF_GetMessage](#) and remove them from their queue by sending a [SIF_Ack](#) message to the ZIS. [SIF_Ack](#) is also sent by Pull-Mode Agents to invoke and end Selective Message Blocking (SMB).

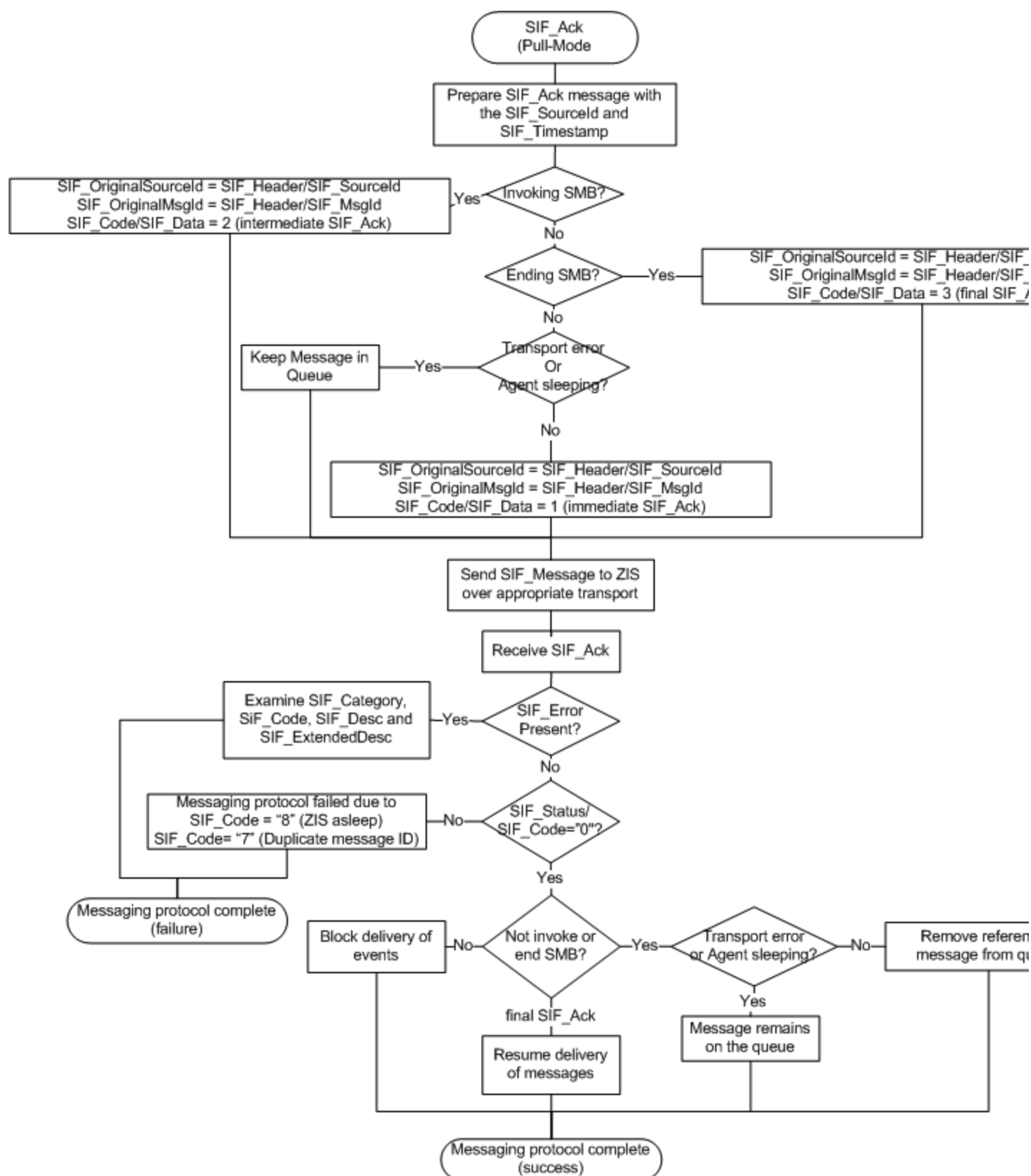


Figure 4.1.1.18-1: SIF_Ack (Pull-Mode) Agent Message Protocol

Step	Process	Flow Control
1	Prepare a <code>SIF_Message/SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply.	If your Agent is invoking SMB, go to Step 3. If your Agent is ending SMB, go to Step 4.

Step	Process	Flow Control
2	From the message being acknowledged/the message to be removed from the queue, place the <code>SIF_Header/SIF_SourceId</code> value into <code>SIF_OriginalSourceId</code> and place the <code>SIF_Header/SIF_MsgId</code> value into <code>SIF_OriginalMsgId</code> . Place 1 (immediate <code>SIF_Ack</code>) into <code>SIF_Code/SIF_Data</code> or an appropriate error description in <code>SIF_Error</code> . If your Agent indicates a transport error or places 8 (receiver is sleeping) into <code>SIF_Code/SIF_Data</code> , the message will be acknowledged but remain in your Agent's queue.	Send <code>SIF_Message</code> to ZIS over appropriate transport. Go to Step 5.
3	From the <code>SIF_Event</code> being blocked, place the <code>SIF_Header/SIF_SourceId</code> value into <code>SIF_OriginalSourceId</code> and place the <code>SIF_Header/SIF_MsgId</code> value into <code>SIF_OriginalMsgId</code> . Place 2 (intermediate <code>SIF_Ack</code>) into <code>SIF_Code/SIF_Data</code> .	Send <code>SIF_Message</code> to ZIS over appropriate transport. Go to Step 5.
4	From the <code>SIF_Event</code> being unblocked, place the <code>SIF_Header/SIF_SourceId</code> value into <code>SIF_OriginalSourceId</code> and place the <code>SIF_Header/SIF_MsgId</code> value into <code>SIF_OriginalMsgId</code> . Place 3 (final <code>SIF_Ack</code>) into <code>SIF_Code/SIF_Data</code> .	Send <code>SIF_Message</code> to ZIS over appropriate transport.
5	Receive <code>SIF_Ack</code> in response. Is <code>SIF_Error</code> present?	If yes, go to Step 9.
6	Is <code>SIF_Status/SIF_Code</code> 0?	If no, go to Step 8.
7	<p>If your Agent did not invoke or end SMB for a <code>SIF_Event</code>, the referenced message has been removed from your Agent's queue, unless your agent indicated a transport error or that it was sleeping (in which case the message has been acknowledged but remains in your Agent's queue).</p> <p>If your Agent invoked SMB by sending an intermediate <code>SIF_Ack</code>, delivery of events is blocked until your Agent removes the <code>SIF_Event</code> from its queue by sending a final <code>SIF_Ack</code>. Your Agent will continue to receive <code>SIF_ResponseS</code> and <code>SIF_RequestS</code>.</p> <p>If your Agent ended SMB by sending a final <code>SIF_Ack</code>, the ZIS has removed the blocked event from your Agent's queue and resumes delivery of events to your Agent.</p>	Messaging protocol complete (success).
8	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate <code>SIF_MsgId</code>).	Messaging protocol complete (failure).
9	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included.	Messaging protocol complete (failure).

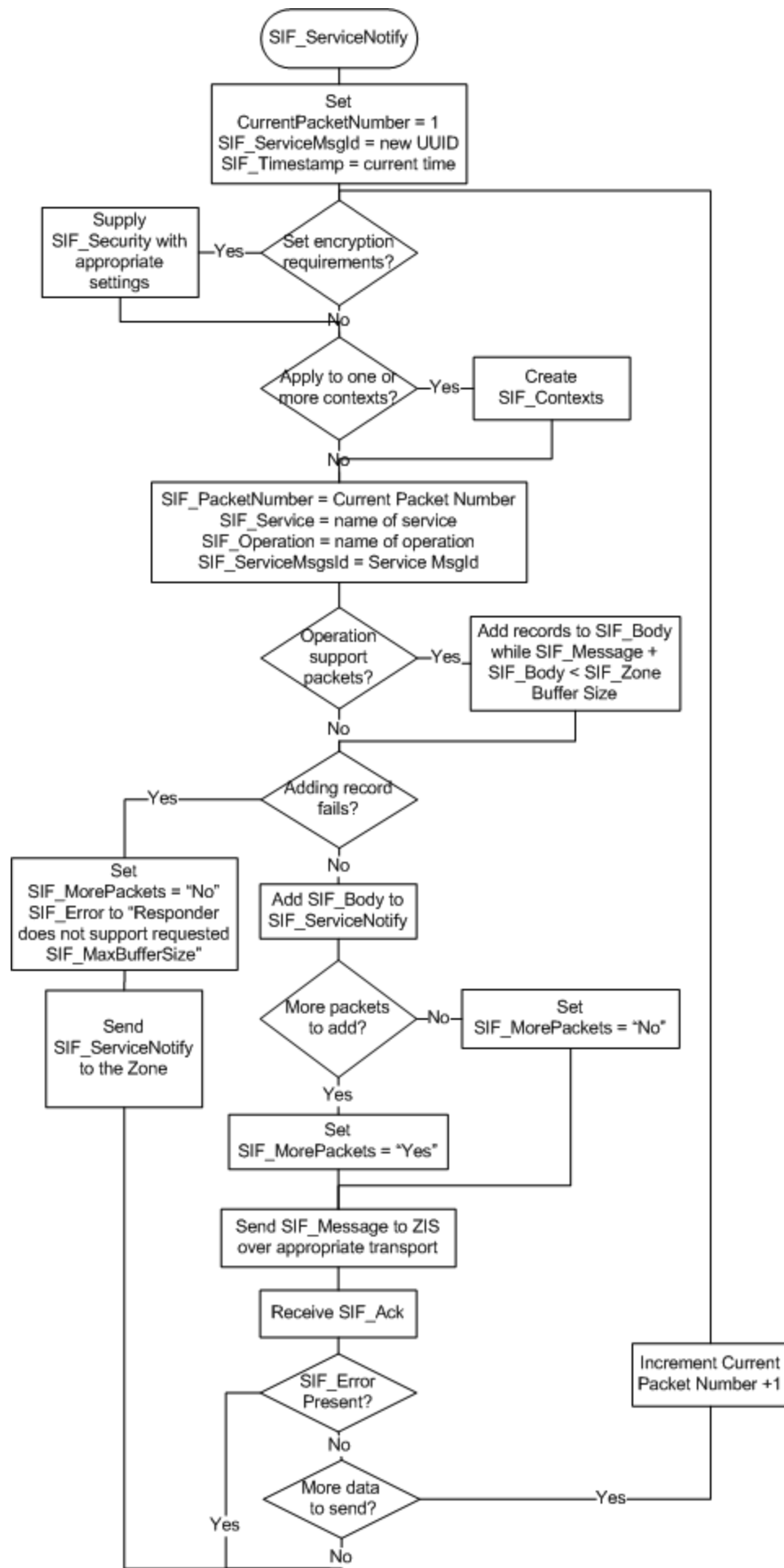
Table 4.1.1.18-1: *SIF_Ack Protocol (Pull-Mode)*

4.1.1.19 SIF_ServiceNotify

`SIF_ServiceNotify` is a message definition used to deliver service notification messages.

When a SIF Zone service wishes to emit a notification message defined by the SIF Zone Service definition to interested subscribers, the Agent that provides the service publishes the corresponding `SIF_ServiceNotify` message to the Zone. Upon successful delivery of a `SIF_ServiceNotify` to the ZIS, the ZIS places the event in the queue for any Agents subscribed to events for the service operations, including the agent that provides the service if the Agent is a subscriber to the notification.

`SIF_ServiceNotify` messages that relate to a SIF Zone service may only be transmitted by the agent that is registered in the zone as the default provider of the service. Unlike Event messages, `SIF_ServiceNotify` messages may be packetized. This means a subscribing client must support the demultiplexing of simultaneously arriving service packets from multiple Zone Service Notification messages.



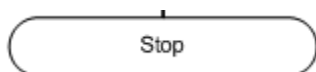


Figure 4.1.1.19-1: SIF_ServiceNotify Agent Message Protocol

Step	Process	Flow Control
1	<p>Initialize Current Packet Number to 1.</p> <p>Initialize ServiceMsgId to a new GUID. The SIF_ServiceMsgId MUST be the same for all packets.</p>	
2	<p>Prepare a new SIF_ServiceNotify message.</p> <p>Initialize the SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp.</p> <p>If the agent would like to indicate minimum encryption and/or authentication requirements for agents receiving this SIF_ServiceNotify, supply SIF_Security with the appropriate settings. Use an equally secure channel when communicating with the Zone, if desired.</p> <p>If this SIF_ServiceNotify specifically applies to one or more contexts, place them in SIF_Contexts; if omitted, the context is SIF_Default.</p> <p>Set SIF_PacketNumber to the Current Packet Number.</p> <p>Set SIF_Service to the name of the SIF Zone Service.</p> <p>Set SIF_Operation to the name of the operation.</p> <p>Set SIF_ServiceMsgId to the ServiceMsgId created in Step 1.</p>	
3	<p>Initialize SIF Zone Service operation SIF_Body and set the appropriate values for the operation call.</p> <p>If the operation SIF_Body supports packets, add records to the SIF_Body while the SIF_Message + SIF_Body is less than either the default SIF Zone Service buffer size or the stated buffer size within the SIF Zone Service documentation. If a record cannot be added under the maximum buffer size abort processing the operation.</p> <p>Add the SIF_Body to the SIF_ServiceNotify.</p>	If a record could not be added go to step 8.
4	If all data records has been added to the SIF_Body set SIF_MorePackets to No. If there is more data to be added in a new SIF_ServiceNotify message set SIF_MorePackets to Yes.	
5	Send SIF_Message/SIF_ServiceNotify to Zone over appropriate communication channel.	If Zone returns SIF_Ack/SIF_Error go to step 9
6	If more data to send increment Current Packet Number +1 and go to step 2	Go to step 2 if more data to send.
7	Processing is complete if no more data left to send.	Stop

Step	Process	Flow Control
8	<p>If a record could not be added to the SIF_ServiceNotify:</p> <p>Set SIF_MorePackets to No.</p> <p>Create a new SIF_Error with the SIF_Error/SIF_Code and SIF_Error/SIF_Desc set to "Responder does not support requested SIF_MaxBufferSize."</p> <p>Send the SIF_ServiceNotify to the Zone. If the first SIF_ServiceNotify packet was not sent, the agent may not have to send the error to the Zone. It may abort the SIF_ServiceNotify.</p> <p>The agent should log the error.</p>	Stop
9	Processing terminated by the Zone.	Stop

Table 4.1.1.19-1: SIF_ServiceNotify Protocol

4.1.1.20 SIF_ServiceInput

This message is used to invoke a method that is exposed by a SIF Zone Service.

An Agent can invoke an operation on a service published by another Agent at any time by sending a SIF_ServiceInput message.

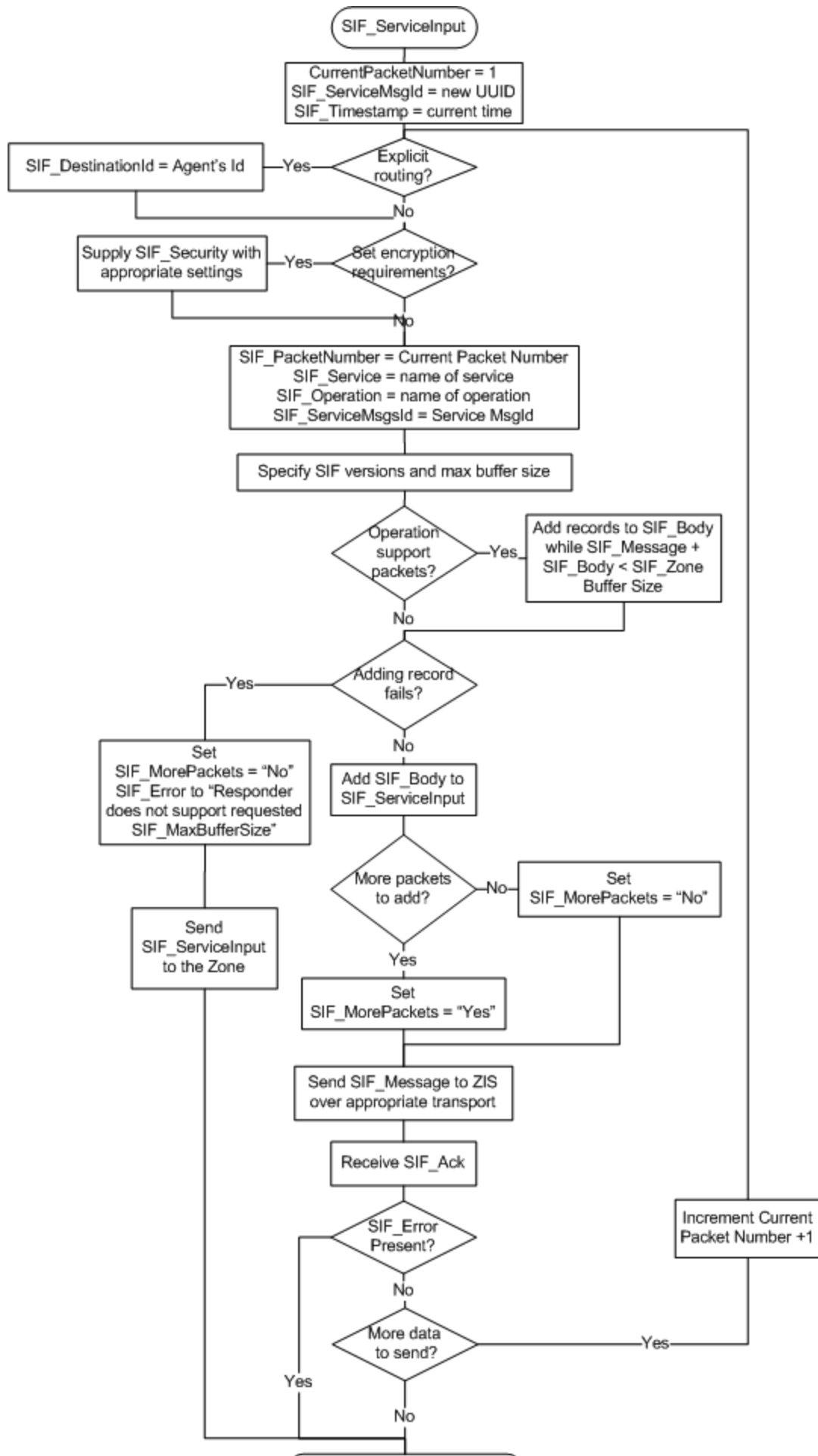




Figure 4.1.1.20-1: SIF_ServiceInput Agent Message Protocol

Step	Process	Flow Control
1	<p>Initialize Current Packet Number to 1</p> <p>Initialize ServiceMsgId to a new GUID. The SIF_ServiceMsgId MUST be the same for all packets.</p>	
2	<p>Prepare a new SIF_ServiceInput message.</p> <p>Initialize the SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp.</p> <p>If your Agent would like to explicitly route this service operation to a given Agent, specify the Agent's Id in SIF_DestinationId.</p> <p>If the agent would like to indicate minimum encryption and/or authentication requirements for agents receiving this SIF_ServiceNotify, supply SIF_Security with the appropriate settings. Use an equally secure channel when communicating with the Zone, if desired.</p> <p>Since a SIF_ServiceInput applies only to the default context, it is not necessary to specify a value for SIF_Contexts.</p> <p>Set SIF_PacketNumber to the Current Packet Number.</p> <p>Set SIF_Service to the name of the SIF Zone Service.</p> <p>Set SIF_Operation to the name of the operation.</p> <p>Set SIF_ServiceMsgId to the ServiceMsgId created in Step 1.</p> <p>Specify the SIF versions the responder may choose from when returning data in SIF_Version. Each version specified MUST be registered in the ZIS as supported by your Agent. It is RECOMMENDED to use 2.*.</p> <p>Specify the maximum buffer size the Responder must respect when sending SIF_ServiceOutput packets; this MUST be less than or equal to the SIF_MaxBufferSize with which your Agent registered with the Zone.</p>	
3	<p>Initialize SIF Zone Service operation SIF_Body and set the appropriate values for the operation call.</p> <p>If the operation SIF_Body supports packets, add records to the SIF_Body while the SIF_Message + SIF_Body is less than either the default SIF Zone Service buffer size or the stated buffer size within the SIF Zone Service documentation. If a record cannot be added under the maximum buffer size, abort processing the operation.</p> <p>Add the SIF_Body to the SIF_ServiceInput.</p>	If a record could not be added go to step 8.
4	If all data records has been added to the SIF_Body set SIF_MorePackets to No. If there is more data to be added in a new SIF_ServiceInput message set SIF_MorePackets to Yes.	
5	Send SIF_Message/SIF_ServiceInput to Zone over appropriate communication channel.	If Zone returns SIF_Ack/SIF_Error go to step 9.
6	If more data to send increment Current Packet Number +1 and go to step 2.	Go to step 2 if more data to send.
7	Processing is complete if no more data left to send.	Stop

Step	Process	Flow Control
8	<p>If a record could not be added to the SIF_ServiceInput:</p> <p>Set SIF_MorePackets to No.</p> <p>Create a new SIF_Error with the SIF_Error/SIF_Code and SIF_Error/SIF_Desc set to "Responder does not support requested SIF_MaxBufferSize."</p> <p>Send the SIF_ServiceNotify to the Zone. If the first SIF_ServiceInput packet was not sent, the agent may not have to send the error to the Zone. It may abort the SIF_ServiceInput.</p> <p>The agent should log the error.</p>	Stop
9	Processing terminated by the Zone.	Stop

Table 4.1.1.20-1: SIF_ServiceInput Protocol

4.1.1.21 SIF_CancelServiceInputs

Agents can request that a ZIS cancel SIF_ServiceInputs pending or in process, by sending a list of SIF_ServiceInMsgIds in a SIF_CancelServiceInputs message. If an Agent abandons or restarts a process choreography which issued SIF_ServiceInputs, whether or not the SIF_ServiceOutput response stream has started, it is **RECOMMENDED** that the Agent send one or more SIF_CancelServiceInputs messages to the ZIS. Supporting such process choreographies can place a heavy load on responding Agents, and cancelling requests may spare Zone resources. Cancelling of responses can also reduce the number of Zone Service output packets the receiving/cancelling agent needs to process and discard.

If the cancelling Agent wishes to receive a "final" SIF_ServiceOutput from the ZIS for each cancelled message, it can specify Standard in SIF_CancelServiceInputs/SIF_NotificationType. If the cancelling Agent does not desire or require "final" SIF_ServiceOutputs, the Agent can specify "None" in SIF_NotificationType.

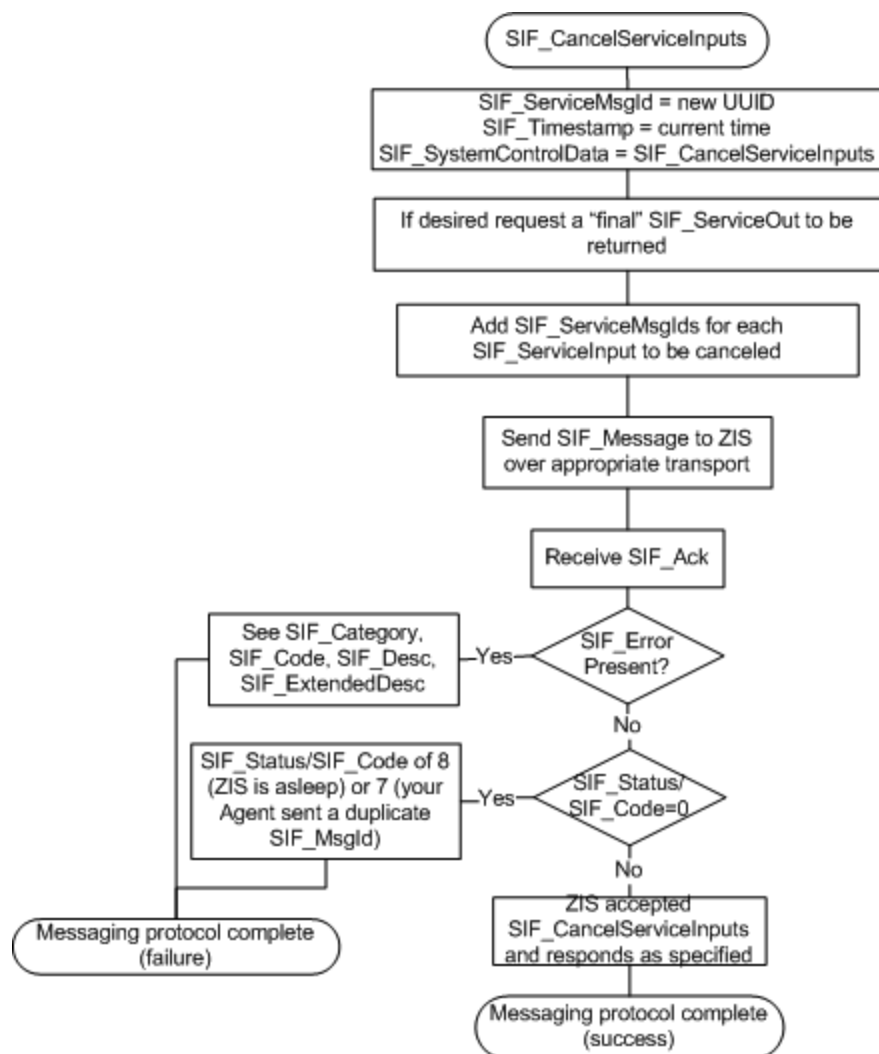


Figure 4.1.1.21-1: SIF_CancelServiceInputs Agent Message Protocol

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId, and the current time in SIF_Timestamp; other SIF_Header elements do not apply. Add a SIF_CancelServiceInputs element in SIF_SystemControlData.	
2	Specify Standard in NotificationType if your Agent desires or requires a "final" SIF_ServiceOut be returned by the ZIS for each cancelled message (SIF_ServiceOut/SIF_MorePackets = No). Otherwise specify None.	
3	Add a SIF_ServiceMsgIds element and add a child SIF_ServiceMsgId element for each SIF_ServiceInput that the Agent wishes to cancel.	Send SIF_Message to ZIS over appropriate transport.
4	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 8.
5	Is SIF_Status/SIF_Code 0?	If no, go to Step 7.
6	The ZIS has accepted the SIF_CancelServiceInputs message. Your Agent will receive or not receive "final" SIF_ServiceOutputs per the specified NotificationType.	Messaging protocol complete (success).
7	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
8	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code, and examine SIF_Desc and SIF_ExtendedDesc, if included.	Messaging protocol complete (failure).

Table 4.1.1.21-1: SIF_CancelServiceInputs Protocol

4.1.2 Agent Message Handling Protocols

This section documents how Agents should respond to incoming messages, and the resulting post-conditions upon success or failure, along with any necessary steps to take.

Note that in handling any `SIF_Message`, an Agent may return a `SIF_Ack` with `SIF_Status/SIF_Code` 8 (receiver is sleeping) or 7 (already have this `SIF_MsgId` from you) if a duplicate message is detected. These responses are typically omitted from the handling protocols below.

4.1.2.1 SIF_Message

Upon receipt of a generic message from the ZIS, in most cases it may be safely assumed that the message XML is well-formed, and perhaps even valid, but the Agent should take the following steps to determine whether the XML is well-formed, optionally validate the message, and check that the message is of a valid type before handing the message off to the respective message handling protocol below.

Step	Process	Flow Control
1	If your transport layer implementation rejects XML that is not well-formed and optionally that is invalid, go to Step 3, 5, 7 or 9 depending on the extent of that implementation. Otherwise, is the XML message well-formed?	If yes, go to step 3.
2	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. If your Agent can scan the incoming message as UTF-8 encoded text to locate <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> , these values can be placed in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Otherwise include these elements with empty values, including an <code>xsi:nil</code> attribute value of true on <code>SIF_OriginalMsgId</code> . Include a <code>SIF_Error</code> element with a <code>SIF_Category</code> of 1 (XML Validation) and a <code>SIF_Code</code> of 2 (message is not well-formed).	Go to step 12.
3	Is the root element of the message unprefixed with a local name of <code>SIF_Message</code> ?	If yes, go to step 5.
4	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Error</code> element with a <code>SIF_Category</code> of 1 (XML Validation) and a <code>SIF_Code</code> of 3 (generic validation error).	Go to step 12.
5	Is the namespace for <code>SIF_Message</code> a namespace of a major version of SIF your Agent supports? Is <code>SIF_Message/@Version</code> present with a value that your Agent supports? (If omitted, interpret <code>SIF_Message/@Version</code> as 1.1.)	If yes, go to step 7.
6	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Error</code> element with a <code>SIF_Category</code> of 12 (Generic Message Handling) and a <code>SIF_Code</code> of 3 (version not supported).	Go to step 12.
7	If your Agent does not validate messages, go to step 9. Otherwise choose a validation schema based on the value of <code>SIF_Message/@Version</code> . Does the message validate?	If yes, go to step 9.
8	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Error</code> element with a <code>SIF_Category</code> of 1 (XML Validation) and an appropriate <code>SIF_Code</code> from the corresponding choices in Error Codes .	Go to step 12.
9	If the namespace for <code>SIF_Message</code> is for a previous major version of SIF, handle according to the specification for <code>SIF_Message/@Version</code> . Otherwise, is the message type (the child element of <code>SIF_Message</code>) <code>SIF_Event</code> , <code>SIF_Request</code> , <code>SIF_Response</code> , <code>SIF_Ping</code> (Push-mode only), <code>SIF_Sleep</code> (Push-mode only), <code>SIF_Wakeup</code> (Push-mode only), or <code>SIF_CancelRequests</code> (Push-mode only and your Agent chooses to support this optional message)?	If yes, go to step 11.

Step	Process	Flow Control
10	Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId , respectively. Include a SIF_Error element with a SIF_Category of 12 (Generic Message Handling) and a SIF_Code of 2 (message not supported).	Go to step 12.
11	Process per the corresponding message handling protocol below.	Message handling is complete.
12	If your Agent is a Push-mode Agent, return the SIF_Ack to the ZIS. If your Agent is a Pull-mode Agent send the SIF_Ack to the ZIS per SIF_Ack (Pull-Mode) above.	Message handling is complete.

Table 4.1.2.1-1: *SIF_Message Handling*

4.1.2.2 SIF_Event

A ZIS places a [SIF_Event](#) in your Agent's queue when an event occurs in a Zone Context with regard to an object for which your agent has subscribed to receive events. A [SIF_Event](#) is delivered when it is the next message pending delivery in the queue.

An event may apply to one or more contexts; these are listed in [SIF_Header/SIF_Contexts](#). If [SIF_Contexts](#) is not present, the context for the event is [SIF_Default](#). The type of event is specified in [SIF_EventObject/@Action](#), the corresponding data object is in [SIF_EventObject](#). A [Change](#) or [Delete](#) event may contain a partial object, but it must include the necessary attribute(s) and element(s) to uniquely identify the object being changed or deleted. These keys/identifiers are typically communicated in the root attributes of an object.

Step	Process	Flow Control
1	Does your Agent invoke Selective Message Blocking (SMB) for all events, or does this event indicate to your Agent that it will invoke SMB?	If no, go to Step 3.
2	<p>Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId, respectively. Place 2 (intermediate SIF_Ack) in SIF_Status/SIF_Code.</p> <p>If your Agent is a Push-Mode Agent, return the SIF_Ack to the ZIS and commence sending the necessary requests as described in the SIF_Request protocol above to complete processing of the event per your Agent's business rules. When complete or if an error occurs, end SMB as described in the SIF_Ack (Push-Mode) protocol above.</p> <p>If your Agent is a Pull-Mode Agent, send the SIF_Ack to the ZIS per SIF_Ack (Pull-Mode) above and commence sending the necessary requests as described in the SIF_Request protocol above to complete processing of the event per your Agent's business rules. When complete or if an error occurs, end SMB as described in the SIF_Ack (Pull-Mode) protocol above.</p> <p>If an error occurs, it is RECOMMENDED that your Agent publish a SIF_LogEntry Add event.</p>	Message handling complete.

Step	Process	Flow Control
3	<p>If your Agent is a Pull-Mode Agent, process the event per your Agent's business rules. When complete or if an error occurs, acknowledge the message and remove it from your Agent's queue per <code>SIF_Ack</code> (Pull-Mode) above.</p> <p>If your Agent is a Push-Mode Agent, it has one of two options: process the event, then acknowledge it; or acknowledge the event, then process it. The advantage of first processing the event is the ability to return a descriptive error, if necessary, to the ZIS when acknowledging the message. The disadvantage of first processing is that if the processing is long running, the connection from the ZIS to your Agent may time out, which will lead to the event being redelivered to your Agent in another delivery attempt, to possibly run into another time-out. To avoid the latter, it is RECOMMENDED that your Push-Mode Agent first acknowledge the event, then process it, unless event processing is known to always occur within a reasonable amount of time. Agents that first acknowledge then process SHOULD persist the event locally until processing is complete, as the event is removed from your Agent's queue upon successful acknowledgement, otherwise the event will be lost in the case of an application or system error that affects your Agent's ability to complete processing of the event.</p> <p>Choose an option and process the event according to your Agent's business rules. When acknowledging: Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code>, your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code>; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code>, respectively. Place 1 (immediate <code>SIF_Ack</code>) in <code>SIF_Status/SIF_Data</code> in the case of successful processing, and return the <code>SIF_Ack</code> to the ZIS. If an error has occurred, include a <code>SIF_Error</code> element with an appropriate <code>SIF_Category</code> and <code>SIF_Code</code> and describe the error as needed in <code>SIF_Desc</code> and optionally <code>SIF_ExtendedDesc</code>. Note that indicating a transport error will not remove the message from your Agent's queue, only acknowledge it. The same action can be accomplished indicating 8 (receiver is sleeping) in <code>SIF_Status/SIF_Code</code>.</p> <p>If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a <code>SIF_LogEntry Add</code> event.</p>	Message handling complete

Table 4.1.2.2-1: *SIF_Event Handling*

4.1.2.3 SIF_Request

A ZIS places a `SIF_Request` in your Agent's queue when an Agent sends a request directly to your Agent, or when an Agent sends a request without a `SIF_DestinationId` and your agent is registered as the Provider for the object requested in `SIF_Query`, or in the case of `SIF_ExtendedQuery` when your agent is registered as the Provider of the object specified by the Requester in `SIF_ExtendedQuery/SIF_DestinationProvider` or `SIF_ExtendedQuery/SIF_From/@ObjectName`. The ZIS will not send your Agent a `SIF_ExtendedQuery` unless your Agent has registered its support for that query type using `SIF_Provide` or `SIF_Provision`. A `SIF_Request` is delivered when it is the next message pending delivery in your Agent's queue.

Any error that occurs while generating `SIF_Responses` during `SIF_Request` handling **MUST** be sent to the Requester with `SIF_MorePackets` set to No, at which point the response stream ends.

Step	Process	Flow Control
1	Examine <code>SIF_Header/SIF_Contexts</code> to determine the context for the request; if none is specified, the context is <code>SIF_Default</code> .	Go to Step 3 if the context is supported.
2	<p>Prepare a <code>SIF_Response</code> message with a copy of <code>SIF_Contexts</code>, <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.</p> <p>Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate General Message Handling and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the requested context is not supported.</p> <p>Add <code>SIF_PacketNumber</code> with a value of 1 and set <code>SIF_MorePackets</code> to No.</p> <p>Send the <code>SIF_Response</code> to the original requester and acknowledge the error to the ZIS.</p>	Message handling complete.

Step	Process	Flow Control
3	Examine the <code>SIF_Version</code> element or elements specified in the <code>SIF_Request</code> message. If more than one version is supported, select the highest version number supported.	Go to Step 5 if a version is supported.
4	<p>Prepare a <code>SIF_Response</code> message with a copy of <code>SIF_Contexts</code>, <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.</p> <p>Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the requested <code>SIF_Versions</code> are not supported.</p> <p>Add <code>SIF_PacketNumber</code> with a value of 1 and set <code>SIF_MorePackets</code> to No.</p> <p>Send the <code>SIF_Response</code> to the original requester and acknowledge the error to the ZIS.</p>	Message handling complete.
5	Examine the <code>SIF_MaxBufferSize</code> specified in the <code>SIF_Request</code> message.	Go to Step 7 if it is greater than or equal to the minimum buffer size your Agent can support. (The buffer size of individual packets will be handled below).
6	<p>Using the SIF version selected in Step 1, prepare a <code>SIF_Response</code> message with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.</p> <p>Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the <code>SIF_MaxBufferSize</code> cannot be honored.</p> <p>Add <code>SIF_PacketNumber</code> with a value of 1 and set <code>SIF_MorePackets</code> to No.</p> <p>Send the <code>SIF_Response</code> to the original requester and acknowledge the error to the ZIS.</p>	Message handling complete.
7	Is <code>SIF_ExtendedQuery</code> specified?	If yes, go to Step 12.
8	The query type is <code>SIF_Query</code> . Examine the object name being queried in <code>SIF_QueryObject/@ObjectName</code> .	Go to Step 10 if the object is supported.
9	<p>Prepare a <code>SIF_Response</code> message using the version chosen in Step 1 with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.</p> <p>Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the object is not supported.</p> <p>Add <code>SIF_PacketNumber</code> with a value of 1 and set <code>SIF_MorePackets</code> to No.</p> <p>Send the <code>SIF_Response</code> to the original requester and acknowledge the error to the ZIS.</p>	Stop processing the message.
10	Examine the query represented, if any, by <code>SIF_ConditionGroup</code> , or <code>SIF_Example</code> in the case of objects that support query-by-example, and determine whether it is supported.	Go to Step 14 if neither <code>SIF_ConditionGroup</code> nor <code>SIF_Example</code> is present, or if the query represented by <code>SIF_ConditionGroup</code> or <code>SIF_Example</code> is supported.

Step	Process	Flow Control
11	<p>Prepare a <code>SIF_Response</code> message with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.</p> <p>Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the query is not supported.</p> <p>Add <code>SIF_PacketNumber</code> with a value of 1 and set <code>SIF_MorePackets</code> to No.</p> <p>Send the <code>SIF_Response</code> to the original requester and acknowledge the error to the ZIS.</p>	Stop processing the message.
12	Examine the query represented by <code>SIF_ExtendedQuery</code> , and determine whether it is supported.	Go to Step 14 if the query is supported.
13	<p>Prepare a <code>SIF_Response</code> message with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.</p> <p>Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the query is not supported.</p> <p>Add <code>SIF_PacketNumber</code> with a value of 1 and set <code>SIF_MorePackets</code> to No.</p> <p>Send the <code>SIF_Response</code> to the original requester and acknowledge the error to the ZIS.</p>	Stop processing the message.
14	<p>Note that Push-Mode Agents should acknowledge receipt of the <code>SIF_Request</code> as response generation is typically a long-running operation that will typically lead to HTTP time-outs. As the request will be removed from the Agent's queue, it is RECOMMENDED that the Push-Mode Agent persist the request and its <code>SIF_PacketNumber</code> while generating responses, in case of an application or system failure that prevents it from completing the request processing; upon restarting, the Agent can end the response stream with a <code>SIF_Error</code>, <code>SIF_PacketNumber</code> set to the last successfully transmitted <code>SIF_PacketNumber + 1</code> and <code>SIF_MorePackets</code> set to No.</p> <p>If a Push-Mode Agent elects to successfully acknowledge the request before processing, it can do so. Otherwise it should acknowledge receipt of the request upon completion of response generation.</p> <p>Pull-Mode Agents can choose to acknowledge receipt of the request here or at the end of response generation.</p> <p>Initialize packet counter to 1.</p>	
15	Prepare a <code>SIF_Response</code> message with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message. When handling <code>SIF_ExtendedQuery</code> , copy the requested columns into <code>SIF_ExtendedQueryResults/SIF_ColumnHeaders</code> .	
16	Add one or more of the matching objects into <code>SIF_ObjectData</code> , for <code>SIF_Query</code> , or rows into <code>SIF_ExtendedQueryResult</code> , for <code>SIF_ExtendedQuery</code> , until no more will fit within the specified buffer size. If no objects or rows will fit within <code>SIF_MaxBufferSize</code> , go to Step 15 with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that <code>SIF_MaxBufferSize</code> cannot be honored. Otherwise, note that only requested columns are returned when processing <code>SIF_ExtendedQuery</code> . When processing <code>SIF_Query</code> , if the requester specified only certain elements be returned, that the Responder needs to return only those elements and their parent elements and attributes. Not supporting a requested element/attribute does not exclude the object from the response stream; include the parent elements/attributes of any missing elements, including the object itself.	If no errors occur in retrieving/adding matching objects, go to step 18.
17	Set <code>SIF_PacketNumber</code> to the current packet counter and <code>SIF_MorePackets</code> to No. Add an appropriate <code>SIF_Error</code> element to the <code>SIF_Response</code> and send the <code>SIF_Response</code> to the ZIS.	Go to Step 21.

Step	Process	Flow Control
18	Set <code>SIF_PacketNumber</code> to the current packet counter value and set <code>SIF_MorePackets</code> appropriately. Send the <code>SIF_Response</code> to the ZIS.	
19	Examine the <code>SIF_Ack</code> returned by the ZIS.	If an error occurred, stop processing the <code>SIF_Request</code> message. Go to Step 21.
20	Determine if more objects or rows match the specified conditions.	If yes, increment the packet counter and go to Step 15; otherwise, go to Step 21.
21	If your Agent has not yet acknowledged receipt of the incoming request, acknowledge successful receipt of the request, or return a descriptive error to the ZIS.	Message handling complete.

Table 4.1.2.3-1: *SIF_Request Handling*

4.1.2.4 SIF_Response

A ZIS places a `SIF_Response` in your Agent's queue when a responder sends a response packet to your Agent per a `SIF_Request` previously sent by your Agent. It is delivered when it is the next message available for delivery to your Agent.

Step	Process	Flow Control
1	<code>SIF_RequestMsgId</code> indicates which of your <code>SIF_Requests</code> this packet is in response to. Is <code>SIF_Error</code> present?	If no, go to Step 3.
2	<p>The Responder's handling of your Agent's <code>SIF_Request</code> has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code>, and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code>, if included. This is the last packet your Agent will receive associated with that request.</p> <p>If your Agent is a Pull-Mode Agent, acknowledge the message per <code>SIF_Ack</code> (Pull-Mode) above.</p> <p>If your Agent is a Push-Mode Agent: Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code>, your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code>; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code>, respectively. Place 1 (immediate <code>SIF_Ack</code>) in <code>SIF_Status/SIF_Data</code>, and return the <code>SIF_Ack</code> to the ZIS.</p>	Message handling complete. Any resources associated with the request can be released.

Step	Process	Flow Control
3	<p>If your Agent is a Pull-Mode Agent, process the response per your Agent's business rules. When complete or if an error occurs, acknowledge the message and remove it from your Agent's queue per <code>SIF_Ack</code> (Pull-Mode) above.</p> <p>If your Agent is a Push-Mode Agent, it has one of two options: process the response, then acknowledge it; or acknowledge the response, then process it. The advantage of first processing the response is the ability to return a descriptive error, if necessary, to the ZIS when acknowledging the message. The disadvantage of first processing is that if the processing is long running, the connection from the ZIS to your Agent may time out, which will lead to the response being redelivered to your Agent in another delivery attempt, to possibly run into another time-out. To avoid the latter, it is RECOMMENDED that your Push-Mode Agent first acknowledge the response, then process it, unless response processing is known to always occur within a reasonable amount of time. Agents that first acknowledge then process SHOULD persist the response locally until processing is complete, as the response is removed from your Agent's queue upon successful acknowledgement, otherwise the response will be lost in the case of an application or system error that affects your Agent's ability to complete processing of the response.</p> <p>Choose an option and process the response according to your Agent's business rules. When acknowledging: Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code>, your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code>; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code>, respectively. Place 1 (immediate <code>SIF_Ack</code>) in <code>SIF_Status/SIF_Data</code> in the case of successful processing, and return the <code>SIF_Ack</code> to the ZIS. If an error has occurred, include a <code>SIF_Error</code> element with an appropriate <code>SIF_Category</code> and <code>SIF_Code</code> and describe the error as needed in <code>SIF_Desc</code> and optionally <code>SIF_ExtendedDesc</code>. Note that indicating a transport error will not remove the message from your Agent's queue, only acknowledge it. The same action can be accomplished indicating 8 (receiver is sleeping) in <code>SIF_Status/SIF_Code</code>.</p> <p>If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a <code>SIF_LogEntry Add</code> event.</p>	<p>Message handling complete. If <code>SIF_MorePackets</code> is No, this is the last packet associated with the request your Agent will receive; any resources associated with the request can be released.</p>

Table 4.1.2.4-1: *SIF_Response Handling*

4.1.2.5 SIF_Ping (Push-Mode only)

The ZIS is pinging your Agent to see if it is reachable, "awake" and/or processing messages.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. If your Agent is "awake," include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 1 (immediate <code>SIF_Ack</code>). Otherwise you may optionally notify the ZIS that your Agent is asleep by returning a <code>SIF_Code</code> of 8 (receiver is sleeping).	
2	Return the <code>SIF_Ack</code> to the ZIS.	Message processing complete (success).

Table 4.1.2.5-1: *SIF_Ping Handling*

4.1.2.6 SIF_Sleep (Push-Mode only)

The ZIS has changed its state to "asleep" and is either not processing incoming messages or all incoming messages will be acknowledged with a `SIF_Ack/SIF_Status/SIF_Code` value of 8 (receiver is sleeping); delivery of queued messages to your Agent is halted. Your Agent **SHOULD** avoid sending messages to the ZIS until receipt of a `SIF_Wakeup` message, or be prepared to handle transport errors or the aforementioned acknowledgement.

Step	Process	Flow Control
1	Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId , respectively. Include a SIF_Status element with a SIF_Code of 1 (immediate SIF_Ack). Change your Agent's ZIS state to "asleep."	
2	Return the SIF_Ack to the ZIS.	Message processing complete (success).

Table 4.1.2.6-1: *SIF_Sleep Handling*

4.1.2.7 SIF_Wakeup (Push-Mode only)

The ZIS has changed its state to "awake" and is processing incoming messages and delivering queued messages again.

Step	Process	Flow Control
1	Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId , respectively. Include a SIF_Status element with a SIF_Code of 1 (immediate SIF_Ack). Change your Agent's ZIS state to "awake."	
2	Return the SIF_Ack to the ZIS.	Message processing complete (success).

Table 4.1.2.7-1: *SIF_Wakeup Handling*

4.1.2.8 SIF_CancelRequests (Push-Mode only) (optional)

A ZIS is requesting that your Agent cancel processing of one or more [SIF_Request](#) messages. Support for handling of this message is currently optional for Push-Mode Agents. If your Agent does not support [SIF_CancelRequests](#), it returns a Generic Message Handling error upon receipt of the [SIF_SystemControl](#) message, error code "Message not supported," per the

[SIF_Message](#) handling protocol.

Step	Process	Flow Control
1	Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId , your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId , respectively. Include a SIF_Status element with a SIF_Code of 1 (immediate SIF_Ack).	
2	If your Agent is currently preparing SIF_Response packets for any of the SIF_Request messages specified in the SIF_RequestMsgId element(s), stop processing the request(s). If your Agent has any of the specified SIF_Requests queued locally, remove them from the agent local queue.	
3	Return the SIF_Ack to the ZIS.	Message processing complete (success).

Table 4.1.2.8-1: *SIF_CancelRequests Handling*

4.1.2.9 SIF_CancelServiceInputs (Push-Mode only) (optional)

A ZIS is requesting that your Agent cancel processing of one or more [SIF_ServiceInput](#) messages. Support for handling of this message is currently optional for Push-Mode Agents. If your Agent does not support [SIF_CancelServiceInputs](#), it returns a Generic Message Handling error upon receipt of the [SIF_SystemControl](#) message, error code "Message not supported," per the

`SIF_Message` handling protocol.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Agent's Agent Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 1 (immediate <code>SIF_Ack</code>).	
2	If your Agent is currently preparing <code>SIF_ServiceOutput</code> packets for any of the <code>SIF_ServiceInput</code> messages specified in the <code>SIF_ServiceMsgId</code> element(s), stop processing the request(s). If your Agent has any of the specified <code>SIF_ServiceInput</code> queued locally, remove them from the agent local queue.	
3	Return the <code>SIF_Ack</code> to the ZIS.	Message processing complete (success).

Table 4.1.2.9-1: `SIF_CancelServiceInputs` Handling

4.1.2.10 `SIF_ServiceNotify`

`SIF_ServiceNotify` is a message definition used to deliver service events.

A ZIS places a `SIF_ServiceNotify` in your Agent's queue when a service notification event occurs in the zone and your agent has previously provisioned itself as a subscriber to that event. A `SIF_ServiceNotify` is delivered when it is the next message pending delivery in the queue.

A service event may only apply to the `SIF_Default` context. If `SIF_Contexts` is not present, the context for the event is `SIF_Default`. The Service that created the event is specified in the `SIF_Service` element. The name of the notification event is specified in the `SIF_Operation` element.

Note that unlike `SIF_Event` messages, `SIF_ServiceNotify` does not support SMB and can be delivered in more than one packet.

Step	Process	Flow Control
1	<p>If your Agent is a Pull-Mode Agent, process the service event per your Agent's business rules. When complete or if an error occurs, acknowledge the message and remove it from your Agent's queue per SIF_Ack (Pull-Mode) above.</p> <p>If your Agent is a Push-Mode Agent, it has one of two options: process the service event, then acknowledge it; or acknowledge the service event, then process it. The advantage of first processing the event is the ability to return a descriptive error, if necessary, to the ZIS when acknowledging the message. The disadvantage of first processing is that if the processing is long running, the connection from the ZIS to your Agent may time out, which will lead to the event being redelivered to your Agent in another delivery attempt, to possibly run into another time-out. To avoid the latter, it is RECOMMENDED that your Push-Mode Agent first acknowledge the event, then process it, unless event processing is known to always occur within a reasonable amount of time. Agents that first acknowledge then process SHOULD persist the event locally until processing is complete, as the event is removed from your Agent's queue upon successful acknowledgement, otherwise the event will be lost in the case of an application or system error that affects your Agent's ability to complete processing of the service event.</p> <p>Choose an option and process the service event according to your Agent's business rules. When acknowledging: Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId, respectively. Place 1 (immediate SIF_Ack) in SIF_Status/SIF_Data in the case of successful processing, and return the SIF_Ack to the ZIS.</p> <p>If an error has occurred, include a SIF_Error element with an appropriate SIF_Category and SIF_Code and describe the error as needed in SIF_Desc and optionally SIF_ExtendedDesc. Note that indicating a transport error will not remove the message from your Agent's queue, only acknowledge it. The same action can be accomplished indicating 8 (receiver is sleeping) in SIF_Status/SIF_Code.</p> <p>If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a SIF_LogEntry Add event.</p>	Message handling complete.

Table 4.1.2.10-1: SIF_ServiceNotify Handling

4.1.2.11 SIF_ServiceInput

This message is used to invoke a method that is exposed by a SIF Zone Service.

A ZIS places a SIF_ServiceInput in your Agent's queue when an Agent sends a directed service request to your Agent, or when an Agent sends a request without a SIF_DestinationId and your agent is registered as the provider of the service specified in SIF_Operation. A SIF_ServiceInput is delivered when it is the next message pending delivery in your Agent's queue.

Any error that occurs while generating SIF_ServiceOutputs during SIF_ServiceInput handling **MUST** be sent to the Requester with SIF_MorePackets set to No, at which point the response stream ends.

An Agent may wait until all SIF_ServiceInput packets have been received before processing the SIF_ServiceInput. This will impact the type of SIF_Ack returned per packet received.

Step	Process	Flow Control
1	Since a SIF_ServiceInput can apply only to the default context, it is not necessary to specify a value for SIF_Header/SIF_Contexts.	Go to step 3.
2	Specify the maximum buffer size the Zone Service must respect when sending SIF_ServiceOutput packets; this MUST be less than or equal to the SIF_MaxBufferSize with which your Agent registered with the ZIS.	

Step	Process	Flow Control
3	Examine the SIF_Version element or elements specified in the SIF_ServiceInput message. If more than one version is supported, select the highest version number supported. If a wildcard * anywhere in a version was specified, choose the maximum matching version supported by your agent or the version appropriate for the Service operation.	Go to step 5 if the version is supported.
4	Prepare a SIF_ServiceOutput message with SIF_DestinationId set to SIF_SourceId and SIF_ServiceMsgId set to SIF_ServiceMsgId from the SIF_ServiceInput message. Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate registration and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that the version is not supported. Add SIF_PacketNumber with a value of 1 and set SIF_MorePackets to No. Send the SIF_ServiceOutput to the Zone and acknowledge the error to the Zone via a SIF_Ack if a SIF_Ack has not already been returned.	Stop processing the message
5	Examine the SIF_MaxBufferSize specified in the SIF_ServiceInput message if it is greater than the minimum buffer size supported by your agent.	Go to step 7 if the buffer size is greater than the minimum buffer size in your agent.
6	Prepare a SIF_ServiceOutput message with SIF_DestinationId set to SIF_SourceId and SIF_ServiceMsgId set to SIF_ServiceMsgId from the SIF_ServiceInput message. Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate Zone Services and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that the SIF_MaxBufferSize is not supported. Add SIF_PacketNumber with a value of 1 and set SIF_MorePackets to No. Send the SIF_ServiceOutput to the Zone and acknowledge the error to the Zone via a SIF_Ack if a SIF_Ack has not already been returned.	Stop processing the message
7	Examine the SIF_ServiceInput/SIF_Service and SIF_ServiceInput/SIF_Operation if they are supported.	Go to step 9 if they are supported
8	Prepare a SIF_ServiceOutput message with SIF_DestinationId set to SIF_SourceId and SIF_ServiceMsgId set to SIF_ServiceMsgId from the SIF_ServiceInput message. Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate Zone Services and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that the SIF_Service and/or SIF_Operation is invalid. Add SIF_PacketNumber with a value of 1 and set SIF_MorePackets to No. Send the SIF_ServiceOutput to the Zone and acknowledge the error to the Zone via a SIF_Ack if a SIF_Ack has not already been returned.	Stop processing the message
9	Process the service operation accordingly.	If the processing is complete go to step 11.
10	Prepare a SIF_ServiceOutput message with SIF_DestinationId set to SIF_SourceId and SIF_ServiceMsgId set to SIF_ServiceMsgId from the SIF_ServiceInput message. Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate Zone Services and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that the SIF_Service and/or SIF_Operation failed. Add SIF_PacketNumber with a value of 1 and set SIF_MorePackets to No. Send the SIF_ServiceOutput to the Zone and acknowledge the error to the Zone via a SIF_Ack if a SIF_Ack has not already been returned.	Stop processing the message
11	Prepare to return the results	

Step	Process	Flow Control
12	<p>Initialize Current Packet Number to 1</p> <p>Collect sender's SIF_SourceId from SIF_ServiceInput/SIF_Header/SIF_SourceId</p> <p>Collect Maximum Buffer Size from SIF_ServiceInput/SIF_MaxBufferSize</p> <p>Collect SIF_Version(s) from SIF_ServiceInput/SIF_Version</p>	
13	<p>Prepare a new SIF_ServiceOutput message.</p> <p>Initialize the SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp.</p> <p>If the agent would like to indicate minimum encryption and/or authentication requirements for agents receiving this SIF_ServiceNotify, supply SIF_Security with the appropriate settings. Use an equally secure channel when communicating with the Zone, if desired.</p> <p>Since SIF_ServiceOutput can apply to only one context, the value in SIF_Contexts is always SIF_Default.</p> <p>Set SIF_ServiceOutput/SIF_Header/SIF_DestinationId to the value from SIF_ServiceInput/SIF_Header/SIF_SourceId</p> <p>Set SIF_PacketNumber to the Current Packet Number.</p> <p>Set SIF_Service to the name of the SIF Zone Service.</p> <p>Set SIF_Operation to the name of the operation.</p> <p>Set SIF_ServiceMsgId to the SIF_ServiceMsgId of the original SIF_ServiceInput/SIF_ServiceMsgId</p>	
14	<p>Initialize SIF Zone Service operation SIF_Body and set the appropriate values for the operation call.</p> <p>If the operation SIF_Body supports packets add records to the SIF_Body while the SIF_Message + SIF_Body is less than either the default SIF Zone Service buffer size or the stated buffer size within the SIF Zone Service documentation. If a record cannot be added under the maximum buffer size abort processing the operation.</p> <p>Add the SIF_Body to the SIF_ServiceOutput</p>	If a record could not be added go to step 19.
15	If all data records has been added to the SIF_Body set SIF_MorePackets to No. If there is more data to be added in a new SIF_ServiceOutput message set SIF_MorePackets to Yes.	
16	Send SIF_Message/SIF_ServiceOutput to Zone over appropriate communication channel.	If Zone returns SIF_Ack/SIF_Error go to step 20
17	If more data to send increment Current Packet Number +1 and go to step 2	Go to step 13 if more data to send.
18	Processing is complete if no more data left to send.	Stop
19	<p>If a record could not be added to the SIF_ServiceOutput</p> <p>Set SIF_MorePackets to No</p> <p>Create a new SIF_Error with the SIF_Error/SIF_Code and SIF_Error/SIF_Desc set appropriately.</p> <p>Send the SIF_ServiceOutput to the Zone. If the first SIF_ServiceOutput packet was not sent, the agent may not have to send the error to the Zone. It may abort the SIF_ServiceOutput.</p> <p>The agent should log the error.</p>	Stop
20	Processing terminated by the Zone.	Stop

Table 4.1.2.11-1: SIF_ServiceInput Handling

4.1.2.12 SIF_ServiceOutput

A ZIS places a SIF_ServiceOut in your Agent's queue when a responder sends a response packet to your Agent per a SIF_ServiceIn previously sent by your Agent. It is delivered when it is the next message available for delivery to your Agent.

Step	Process	Flow Control
1	SIF_ServiceInMsgId indicates which of your SIF_ServiceIns this packet is in response to. Is SIF_Error present?	If no, go to Step 3.

Step	Process	Flow Control
2	<p>The Responder's handling of your Agent's SIF_ServiceIn has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code, and examine SIF_Desc and SIF_ExtendedDesc, if included. This is the last packet your Agent will receive associated with that request.</p> <p>If your Agent is a Pull-Mode Agent, acknowledge the message per SIF_Ack (Pull-Mode) above.</p> <p>If your Agent is a Push-Mode Agent: Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId, respectively. Place 1 (immediate SIF_Ack) in SIF_Status/SIF_Data, and return the SIF_Ack to the ZIS.</p>	<p>Message handling complete. Any resources associated with the request can be released.</p>
3	<p>If your Agent is a Pull-Mode Agent, process the response per your Agent's business rules. When complete or if an error occurs, acknowledge the message and remove it from your Agent's queue per SIF_Ack (Pull-Mode) above.</p> <p>If your Agent is a Push-Mode Agent, it has one of two options: process the response, then acknowledge it; or acknowledge the response, then process it. The advantage of first processing the response is the ability to return a descriptive error, if necessary, to the ZIS when acknowledging the message. The disadvantage of first processing is that if the processing is long running, the connection from the ZIS to your Agent may time out, which will lead to the response being redelivered to your Agent in another delivery attempt, to possibly run into another time-out. To avoid the latter, it is RECOMMENDED that your Push-Mode Agent first acknowledge the response, then process it, unless response processing is known to always occur within a reasonable amount of time. Agents that first acknowledge then process SHOULD persist the response locally until processing is complete, as the response is removed from your Agent's queue upon successful acknowledgement, otherwise the response will be lost in the case of an application or system error that affects your Agent's ability to complete processing of the response.</p> <p>Choose an option and process the response according to your Agent's business rules. When acknowledging: Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId, your Agent's Agent Id in SIF_SourceId and the current time in SIF_Timestamp; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId, respectively. Place 1 (immediate SIF_Ack) in SIF_Status/SIF_Data in the case of successful processing, and return the SIF_Ack to the ZIS. If an error has occurred, include a SIF_Error element with an appropriate SIF_Category and SIF_Code and describe the error as needed in SIF_Desc and optionally SIF_ExtendedDesc. Note that indicating a transport error will not remove the message from your Agent's queue, only acknowledge it. The same action can be accomplished indicating 8 (receiver is sleeping) in SIF_Status/SIF_Code.</p> <p>If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a SIF_LogEntry Add event.</p>	<p>Message handling complete. If SIF_MorePackets is No, this is the last packet associated with the request your Agent will receive; any resources associated with the request can be released.</p>

Table 4.1.2.12-1: SIF_ServiceOutput Handling

4.2 ZIS Protocols

4.2.1 ZIS Messaging Protocols

This section documents how Zone Integration Servers send individual messages, and the resulting post-conditions upon success or failure, along with any necessary steps to take. These correspond to each of the actions a Zone Integration Server can initiate.

4.2.1.1 SIF_Message Delivery (SIF_Event, SIF_Request, SIF_Response, SIF_ServiceInput, SIF_ServiceOutput, SIF_ServiceNotify to a Push-mode Agent)

A ZIS contacts a Push-Mode Agent to deliver SIF_Event, SIF_Request and SIF_Response messages queued for the Agent. This delivery protocol starts with a check on whether there are messages pending, as the protocol can loop as messages are delivered.

Step	Process	Flow Control
1	Are there messages queued for the Agent?	If yes, go to Step 2. Otherwise messaging protocol complete (success).
2	Is the state of the Agent "asleep?" If yes, the ZIS SHOULD wait until the Agent sends a SIF_Wakeup message or re-registers in Push mode before attempting message delivery. Otherwise the ZIS MUST be prepared to handle transport errors/exceptions and/or the Agent responding with a SIF_Status/SIF_Code of 8 (receiver is sleeping).	If no, go to Step 3. Otherwise messaging protocol complete (success).
3	Has the Agent previously invoked SMB?	If no, go to Step 6.
4	Iterate through the Agent's queue from the message received first to the most recently received message. Stop at the first SIF_Response or SIF_Request in the queue, if one exists.	If one exists, it is the next message to be delivered. Go to Step 7.
5	The only messages queued for the Agent are SIF_Events; try again later, or after a SIF_Response or SIF_Request arrives, or after the Agent has ended SMB by sending a final SIF_Ack.	Messaging protocol complete (no message needs to be delivered).
6	The next message to be delivered is the message received first in the Agent's queue.	
7	Is SIF_Header/SIF_Security present in the SIF_Message with SIF_EncryptionLevel, SIF_AuthenticationLevel, or both?	If no, the message delivery encryption/authentication levels are the minimum encryption/authentication levels set up for the Zone. Go to Step 9.
8	The ZIS MUST guarantee that the minimum encryption and/or authentication levels specified are respected when delivering this message. Use the higher of these and the Zone's minimum encryption and/or authentication levels during message delivery.	
9	If a connection is already open to the Push-Mode Agent from a previously delivered message, are the encryption and authentication levels greater than or equal to those needed for the delivery of this message?	If there is no connection open, go to Step 11. If there is and the encryption/authentication levels are adequate for delivery, go to Step 15.
10	Attempt to renegotiate the encryption/authentication levels for the connection, or close the connection and attempt to open a new connection with adequate encryption/authentication levels.	Go to Step 12.
11	If the registered transport layer is known to not provide adequate encryption/authentication levels (e.g. SIF HTTP), go to Step 12. Otherwise attempt to open a connection to the Agent with adequate encryption/authentication levels, using the appropriate transport layer.	
12	Was a connection opened or renegotiated with adequate encryption/authentication levels? If no, the message cannot be delivered; remove it from the Agent's queue. It is RECOMMENDED that your ZIS log the error. Your ZIS MUST post a SIF_LogEntry Add event with the appropriate error category and code, containing a copy of the SIF_Header of the queued message. SIF_LogEntry/SIF_Desc MUST contain the SIF_SourceId of the Agent that failed to receive the message.	Go to Step 1 to start delivery of the next queued message, if desired. Otherwise messaging protocol complete (error).
13	Is the agent issuing the notification a provider of the corresponding service?	If yes, go to Step 15
14	Prepare a SIF_Ack/SIF_Error category 14 (SIF Zone Service) code 17 (Not a provider for this service) and send it to the agent improperly issuing the event.	
15	Send the message to the Agent over the connection.	
16	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 26.
17	Is SIF_Status/SIF_Code 1 (immediate SIF_Ack)?	If no, go to Step 19.

Step	Process	Flow Control
18	The Agent has successfully acknowledged receipt of the message; remove it from the Agent's queue.	Go to Step 1 to start delivery of the next queued message, if desired. Otherwise messaging protocol complete (success).
19	Is <code>SIF_Status/SIF_Code 2</code> (intermediate <code>SIF_Ack</code>)?	If no, go to Step 23.
20	The Agent is invoking SMB. Is the delivered message a <code>SIF_Event</code> ?	If yes, go to Step 22.
21	The Agent has violated protocol; remove the message from the Agent's queue. It is RECOMMENDED that your ZIS log the error. Your ZIS MUST post a <code>SIF_LogEntry Add</code> event with the appropriate error category of 13 (SMB Error) and code 2 (SMB can only be invoked for <code>SIF_Event</code>), containing a copy of the <code>SIF_Header</code> of the queued message. <code>SIF_LogEntry/SIF_Desc</code> MUST contain the <code>SIF_SourceId</code> of the Agent that committed the protocol error.	Go to Step 1 to start delivery of the next queued message, if desired. Otherwise messaging protocol complete (error).
22	The Agent has invoked SMB on this <code>SIF_Event</code> . Persist that the Agent has invoked SMB along with the <code>SIF_MsgId</code> of the event. The event stays in the agent's queue as blocked, and all other events are frozen until the Agent eventually ends SMB by sending a final <code>SIF_Ack</code> with this <code>SIF_MsgId</code> in <code>SIF_OriginalMsgId</code> , or by sending a <code>SIF_Wakeup</code> or by re-registering.	Go to Step 1 to start delivery of the next queued message, if desired. Otherwise messaging protocol complete (success).
23	Is <code>SIF_Status/SIF_Code 8</code> (receiver is sleeping)?	If no, go to Step 25.
24	The Agent is asleep. Re-attempt delivery later.	Messaging protocol complete (success).
25	Messaging protocol has failed due to a <code>SIF_Status/SIF_Code</code> of 7 (already have this <code>SIF_MsgId</code>). The ZIS cannot correct this, as the <code>SIF_MsgId</code> originates from an Agent and can't be changed without other repercussions. Remove the message from the Agent's queue. It is RECOMMENDED that your ZIS log the error. Your ZIS MUST post a <code>SIF_LogEntry Add</code> event with the appropriate error category and code, containing a copy of the <code>SIF_Header</code> of the queued message. <code>SIF_LogEntry/SIF_Desc</code> MUST contain the <code>SIF_SourceId</code> of the Agent that did not receive the message.	Go to Step 1 to start delivery of the next queued message, if desired. Otherwise messaging protocol complete (error).
26	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included. If <code>SIF_Category</code> does not indicate a transport error, remove the message from the Agent's queue. Otherwise re-attempt delivery of this message later. It is RECOMMENDED that your ZIS log the error. Your ZIS MAY post a <code>SIF_LogEntry Add</code> event with the appropriate error category and code, containing a copy of the <code>SIF_Header</code> of the queued message. <code>SIF_LogEntry/SIF_Desc</code> MUST contain the <code>SIF_SourceId</code> of the Agent that indicated the error.	Go to Step 1 to start delivery of the next queued message, if desired. Otherwise messaging protocol complete (error).

Table 4.2.1.1-1: *SIF_Message Delivery Protocol*

4.2.1.2 SIF_Ping (to a Push-mode Agent)

A ZIS can "ping" a Push-Mode Agent or check that it's "awake" by sending a

`SIF_Ping` message to the Agent. If the Agent returns a successful acknowledgement, it is awake; the Agent may also reply that it is asleep. As a Push-Mode Agent may be offline completely, Zone Integration Servers should be prepared to handle transport errors directly or wrapped in a `SIF_Ack/SIF_Error` by underlying code.

Step	Process	Flow Control
1	Prepare a <code>SIF_SystemControl</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , the Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place an empty <code>SIF_Ping</code> element in <code>SIF_SystemControlData</code> .	Send <code>SIF_Message</code> to Agent over appropriate transport.
2	Receive <code>SIF_Ack</code> in response. Is <code>SIF_Error</code> present?	If yes, go to Step 8.
3	Is <code>SIF_Status/SIF_Code 1</code> ?	If no, go to Step 5.
4	The Agent is awake.	Messaging protocol complete (success).
5	Is <code>SIF_Status/SIF_Code 8</code> (receiver is sleeping)?	If no, go to Step 7.

Step	Process	Flow Control
6	The Agent is asleep.	Messaging protocol complete (success).
7	Messaging protocol has failed due to a SIF_Status/SIF_Code of 7 (your ZIS sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
8	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.2.1.2-1: *SIF_Ping Protocol*

4.2.1.3 SIF_Sleep (to a Push-mode Agent)

A ZIS can send a

[SIF_Sleep](#) message to a Push-Mode Agent to change its state to "sleeping," indicating that it will either be offline or acknowledging incoming messages with a [SIF_Status/SIF_Code](#) of 8 (receiver is sleeping), and that it will not be delivering messages to the Agent until it "wakes up" by sending a [SIF_Wakeup](#) message.

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Zone Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place an empty SIF_Sleep element in SIF_SystemControlData .	Send SIF_Message to Agent over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 1?	If no, go to Step 5.
4	The Agent has successfully acknowledged your SIF_Sleep and should not be expecting further message delivery until your ZIS sends a SIF_Wakeup .	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (Agent is asleep) or 7 (your ZIS sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.2.1.3-1: *SIF_Sleep Protocol*

4.2.1.4 SIF_Wakeup (to a Push-mode Agent)

A ZIS can send a

[SIF_Wakeup](#) message to a Push-Mode Agent to change its state to "awake;" i.e., that it is ready to process incoming messages and deliver queued messages again.

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Zone Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place an empty SIF_Wakeup element in SIF_SystemControlData .	Send SIF_Message to Agent over appropriate transport.
2	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 6.
3	Is SIF_Status/SIF_Code 1?	If no, go to Step 5.
4	The Agent has successfully acknowledged your "awake" status.	Messaging protocol complete (success).
5	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (Agent is asleep) or 7 (your ZIS sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
6	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	Messaging protocol complete (failure).

Table 4.2.1.4-1: *SIF_Wakeup Protocol*

4.2.1.5 SIF_CancelRequests (to a Push-mode Agent)

A ZIS can send a

SIF_CancelRequests message to a Push-Mode Agent after receiving a **SIF_CancelRequests** messages from another agent, as per the **SIF_CancelRequests message handling protocol**. As support for this message is currently optional for Push-Mode Agents, the ZIS should be prepared to handle a Generic Message Handling error from the Agent upon receipt of the **SIF_SystemControl** message, error code "Message not supported."

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Zone Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place a SIF_CancelRequests element in SIF_SystemControlData .	
2	Place the requests that should be cancelled in SIF_RequestMsgIds/SIF_RequestMsgId . While it is not used by the Push-Mode Agent, set the NotificationType to None .	Send SIF_Message to Agent over appropriate transport.
3	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 7.
4	Is SIF_Status/SIF_Code 1?	If no, go to Step 6.
5	The Agent has successfully acknowledged your SIF_CancelRequests and should have cancelled any corresponding response activity.	Messaging protocol complete (success).
6	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (Agent is asleep) or 7 (your ZIS sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).
7	Messaging protocol has failed due to a SIF_Error condition. See Error Codes with SIF_Category and SIF_Code , and examine SIF_Desc and SIF_ExtendedDesc , if included.	If the SIF_Error is a Generic Message Handling error, error code "Message not supported," go to Step 8. Otherwise messaging protocol complete (failure).
8	The Agent does not support SIF_CancelRequests .	Messaging protocol complete (success).

Table 4.2.1.5-1: SIF_CancelRequests Protocol

4.2.1.6 SIF_CancelServiceInputs (to a Push-mode Agent)

A ZIS can send a

SIF_CancelServiceInputs message to a Push-Mode Agent after receiving a **SIF_CancelServiceInputs** messages from another agent, as per the **SIF_CancelServiceInputs message handling protocol**. As support for this message is currently optional for Push-Mode Agents, the ZIS should be prepared to handle a Generic Message Handling error from the Agent upon receipt of the **SIF_SystemControl** message, error code "Message not supported."

Step	Process	Flow Control
1	Prepare a SIF_Message/SIF_SystemControl message with SIF_Header containing a new GUID in SIF_MsgId , your Zone Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place a SIF_CancelServiceInputs element in SIF_SystemControlData .	
2	Place the requests that should be cancelled in SIF_ServiceMsgIds/SIF_ServiceMsgId . While it is not used by the Push-Mode Agent, set the NotificationType to None .	Send SIF_Message to Agent over appropriate transport.
3	Receive SIF_Ack in response. Is SIF_Error present?	If yes, go to Step 7.
4	Is SIF_Status/SIF_Code 1?	If no, go to Step 6.
5	The Agent has successfully acknowledged your SIF_CancelServiceInputs and should have cancelled any corresponding response activity.	Messaging protocol complete (success).
6	Messaging protocol has failed due to a SIF_Status/SIF_Code of 8 (Agent is asleep) or 7 (your ZIS sent a duplicate SIF_MsgId).	Messaging protocol complete (failure).

Step	Process	Flow Control
7	Messaging protocol has failed due to a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included.	If the <code>SIF_Error</code> is a Generic Message Handling error, error code "Message not supported," go to Step 8. Otherwise messaging protocol complete (failure).
8	The Agent does not support <code>SIF_CancelServiceInputs</code> .	Messaging protocol complete (success).

Table 4.2.1.6-1: `SIF_CancelServiceInputs` Protocol

4.2.2 ZIS Message Handling Protocols

This section documents how Zone Integration Servers should respond to incoming messages, and the resulting post-conditions upon success or failure, along with any necessary steps to take.

Note that in handling any `SIF_Message`, the ZIS can return a `SIF_Ack` with `SIF_Status/SIF_Code` 8 (receiver is sleeping) or 7 (already have this `SIF_MsgId` from you) if a duplicate message is detected. These responses are omitted from the handling protocols below.

4.2.2.1 `SIF_Message`

When a message is received, the ZIS should first validate the XML message. If the message is not `SIF_Register`, the ZIS should determine whether the sender is registered in the zone. If errors are found, a `SIF_Ack` with a `SIF_Error` element should be returned to the caller and no further processing should occur. If no errors are found, message processing proceeds according to message type. Subsequent message processing sections are assured of receiving well-formed and/or valid XML, and all non-`SIF_Register` message processing sections are assured that the agent is indeed registered with the zone.

Step	Process	Flow Control
1	Validate incoming XML message. Message validation is optional. The <code>Version</code> attribute of <code>SIF_Message</code> can be used to indicate the appropriate message definition.	If not performing message validation, go to Step 3 if XML is well-formed. If performing message validation, go to Step 3 if message is well-formed and valid.
2	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. (Note that if XML is not well-formed, or invalid and the well-formed XML is not made available by the XML parser, <code>SIF_SourceId</code> and <code>SIF_MsgId</code> will not be available from the incoming XML message. If this is the case, include <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> in the <code>SIF_Ack</code> as empty elements with <code>xsi:nil</code> set to true as necessary to indicate the current message.) Set <code>SIF_Error/SIF_Category</code> to indicate XML Validation and place the appropriate error code and description in <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> . Place any additional parser information into <code>SIF_Error/SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to caller. If it can be determined the message is a <code>SIF_Response</code> , see SIF_Response Handling below, Step 13, to send an error <code>SIF_Response</code> to the requester.	Stop processing this message.
3	Examine the <code>Version</code> attribute of the message.	If the version is supported, go to Step 5.
4	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to Generic Message Handling, indicating that the message is not supported in <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> . Return the <code>SIF_Ack</code> to the caller. If this message is a <code>SIF_Response</code> , see SIF_Response Handling below, Step 13, to send an error <code>SIF_Response</code> to the requester.	Stop processing this message.
5	Examine message's <code>SIF_Header</code> to retrieve the <code>SIF_SourceId</code> and the message to get the message type. If message type is not <code>SIF_Register</code> , determine if the sender identified by <code>SIF_SourceId</code> is registered.	If message type is <code>SIF_Register</code> or if the sender's <code>SIF_SourceId</code> is registered, go to Step 9.
6	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to Access and Permissions, indicating that the sender is not registered in <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> . Return the <code>SIF_Ack</code> to the caller. If this message is a <code>SIF_Response</code> , see SIF_Response Handling below, Step 13, to send an error <code>SIF_Response</code> to the requester.	Stop processing the message.
7	Is the agent issuing the notification a provider of the corresponding service?	If yes, go to Step 9.

Step	Process	Flow Control
8	Prepare a SIF_Ack/SIF_Error category 14 (SIF Zone Service) code 17 (Not a provider for this service) and send it to the agent improperly issuing the event.	
9	Forward message to the proper handler based on the message type.	

Table 4.2.2.1-1: SIF_Message Handling

4.2.2.2 SIF_Register

Before an agent can participate in a zone, it must register itself in order to provide the data that the ZIS needs to interact with the agent. This process is handled using a SIF_Register message.

Step	Process	Flow Control
1	If ZIS implementation limits SIF_SourceId values in some way, examine SIF_SourceId and determine whether it is valid.	If implementation allows any SIF_SourceId or if the SIF_SourceId is valid, go to Step 3.
2	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Registration and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that SIF_SourceId is invalid. Return the SIF_Ack to the caller.	Stop processing this message.
3	If ZIS implementation requires previous permissions to register, examine SIF_SourceId and determine whether sender is permitted to register.	If implementation allows any sender to register or if sender is permitted to register, go to Step 5.
4	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Access and Permissions and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the lack of permission to register. Return the SIF_Ack to the caller.	Stop processing this message.
5	Examine SIF_Version element(s) and determine if the ZIS can handle the version(s).	Go to Step 7 if the ZIS can handle the SIF version(s) specified by agent.
6	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Registration and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that the ZIS cannot handle SIF messages in a version requested. Place the unsupported version in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	Stop processing this message.
7	Examine SIF_MaxBufferSize and verify that it is greater than or equal to the minimum value for the ZIS.	Go to Step 9 if SIF_MaxBufferSize is large enough.
8	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Registration and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that the SIF_MaxBufferSize is too small to be supported by the ZIS. Return the SIF_Ack to the caller.	Stop processing this message.
9	If the supplied value of SIF_Mode is Push, verify that the SIF_Protocol element is provided and that the protocol information appears sufficient for contacting the agent in Push mode and that the ZIS supports the Accept-Encoding SIF_Protocol/SIF_Property, if specified.	Go to Step 11 if SIF_Mode is Pull or SIF_Protocol information appears valid.
10	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Registration and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that the protocol is not supported, a secure transport is required, or that the ZIS does not support the supplied Accept-Encoding value. Return the SIF_Ack to the caller.	Stop processing this message.
11	Store data from the SIF_Register message into the agent's database profile.	
12	Prepare a SIF_Ack containing a SIF_Status element indicating success, placing the agent's access control permissions in SIF_Status/SIF_Data/SIF_AgentACL. Return the SIF_Ack to the caller.	Stop processing this message.

Table 4.2.2.2-1: SIF_Register Handling

An agent may also send the SIF_Register message when already registered. In this case, the ZIS should re-register the agent in the same manner as defined for initial registration. Any existing provision and subscription entries, as well as any pending messages,

maintained by the ZIS for the agent should remain intact. Upon successful re-registration, any new or updated registration settings for the agent, including push mode protocol information, take effect after the ZIS has returned a successful `SIF_Ack` for the `SIF_Register` message.

4.2.2.3 SIF_Unregister

When an agent is going to be removed from a Zone, the agent must send a `SIF_Unregister` message. When a ZIS receives this message from an agent, it performs those steps—ignoring `SIF_Ack` preparation and delivery—outlined for the `SIF_Unprovide` and `SIF_Unsubscribe` messages for any agent provisions or subscriptions, respectively. The ZIS then discards any messages pending for the agent. The ZIS will also remove any registration information and remove the agent from its list of registered agents.

It is **RECOMMENDED** that the ZIS not remove access control data from its database as a replacement agent may be installed. Keeping the access permissions is optional, however.

Step	Process	Flow Control
1	Examine message and retrieve the <code>SIF_SourceId</code> of the message. The ZIS must remove the agent from its list of registered agents. Perform <code>SIF_Unprovide</code> functionality for any objects the agent is providing. Perform <code>SIF_Unsubscribe</code> functionality for any objects to which the agent is subscribed. Discard any pending messages for the agent.	
2	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Status</code> element indicating success. Return the <code>SIF_Ack</code> to caller.	Stop processing the message.

Table 4.2.2.3-1: `SIF_Unregister` Handling

4.2.2.4 SIF_Provide

An agent makes an object available to be requested by a process called Provision that is represented by the `SIF_Provide` message.

The `SIF_Provide` message can contain provision requests for multiple objects. The ZIS must treat all of the objects as a set; if there is an error with one of the objects then there should be no change to the Providers database.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> .	Go to Step 3.
2	Examine the message to determine whether any more objects are being provided.	Go to Step 11 if there are no further object provisions to process for this message.
3	Retrieve the name of the next object to be provided. If not otherwise performed in initial message validation, check whether the object name is valid (e.g. valid/supported object, not <code>SIF_ZoneStatus</code>).	If object name is valid, go to Step 5.
4	Add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> . Set <code>SIF_Error/SIF_Category</code> to indicate Provision and set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the object is invalid. Place the name of the invalid object in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 14.
5	If no <code>SIF_Context</code> is specified, the context is <code>SIF_Default</code> . Otherwise check that each <code>SIF_Context</code> supplied in <code>SIF_Contexts</code> is supported.	If they are all supported, go to Step 7.
6	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Generic Message Handling. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate a context is not supported. Place the name of the context in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 14.
7	Using the <code>SIF_SourceId</code> , consult the ACL to determine if the sender has the proper access and permissions for this object in each of the specified contexts.	If sender has the proper access and permissions, go to Step 9.
8	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Access and Permissions. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the sender lacks permission to provide this object. Place the name of the object in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 14.
9	Check the Providers database to see if this object has already been provided in the contexts specified.	If the object does not have a provider in the contexts specified, go to Step 11.

Step	Process	Flow Control
10	Is the current provider the same as the <code>SIF_SourceId</code> of this message?	If the provider differs from the <code>SIF_SourceId</code> of this message, go to Step 14. Otherwise go to Step 2.
11	Add a record in the Providers database to indicate that <code>SIF_SourceId</code> is the provider of this object in the given contexts. If an error occurs, add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> .	If an error occurs, go to Step 13; otherwise go to Step 2.
12	Add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> . Set <code>SIF_Error/SIF_Category</code> to indicate Provision and set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the object already has a provider. Place the name of the provider in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 14.
13	Add a <code>SIF_Status</code> element indicating success to the <code>SIF_Ack</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
14	Undo all changes to the Providers database. Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.

Table 4.2.2.4-1: *SIF_Provide Handling*

4.2.2.5 SIF_Unprovide

If an agent wishes to withdraw an object previously provided, the `SIF_Unprovide` message is used.

The `SIF_Unprovide` message can contain multiple objects. The ZIS must treat all of the objects as a set; if there is an error with one of the objects then there should be no change to the Providers database.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> .	Go to Step 3.
2	Examine the message to determine whether any more objects are being unprovided.	Go to Step 7 if there are no further objects to process for this message.
3	Examine the message and retrieve the name of an object to be unprovided. If not otherwise performed in initial message validation, check whether the object name is valid (e.g. valid/supported object, not <code>SIF_ZoneStatus</code>).	Go to Step 5 if the object name is valid.
4	Add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> . Set <code>SIF_Error/SIF_Category</code> to indicate Provision and set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the object is invalid. Place the name of the invalid object in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 10.
5	If no <code>SIF_Context</code> is specified, the context is <code>SIF_Default</code> . Otherwise check that each <code>SIF_Context</code> supplied in <code>SIF_Contexts</code> is supported.	If they are all supported, go to Step 7.
6	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Generic Message Handling. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate a context is not supported. Place the name of the context in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 10.
7	If it exists, remove the records in the Providers database that marks <code>SIF_SourceId</code> as the provider of this object for the given contexts. If an error occurs, add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> .	If an error occurs, go to Step 10.
8	Leave all pending <code>SIF_Requests</code> for the object in the responder's queue, as they may include <code>SIF_Requests</code> routed explicitly to the responder using <code>SIF_DestinationId</code> .	Go to Step 2.
9	Add a <code>SIF_Status</code> element indicating success to the <code>SIF_Ack</code> . Return the <code>SIF_Ack</code> to the caller	Stop processing the message.
10	Undo all changes to the Providers database. Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.

Table 4.2.2.5-1: *SIF_Unprovide Handling*

4.2.2.6 SIF_Subscribe

An agent requests to receive `SIF_Events` for an object by a process called Subscription that is represented by the `SIF_Subscribe` message.

The `SIF_Subscribe` message can contain subscription requests for multiple objects. The ZIS must treat all of the objects as a set, if there is an error with one of the objects then there should be no change to the Subscribers database.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> .	Go to Step 3.
2	Examine the message to determine whether any more subscriptions need to be processed.	Go to Step 9 if there are no further subscriptions to process in this message.
3	Retrieve the name of the next object to be subscribed to. If not otherwise performed in initial message validation, check whether the object name is valid (e.g., valid/supported object with events reported).	If the object name is valid, go to Step 5.
4	Add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> . Set <code>SIF_Error/SIF_Category</code> to indicate Subscription and set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the object is invalid. Place the name of the invalid object in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 12.
5	If no <code>SIF_Context</code> is specified, the context is <code>SIF_Default</code> . Otherwise check that each <code>SIF_Context</code> supplied in <code>SIF_Contexts</code> is supported.	If they are all supported, go to Step 7.
6	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Generic Message Handling. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate a context is not supported. Place the name of the context in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 12.
7	Using the <code>SIF_SourceId</code> , consult the ACL to determine if the sender has the proper access and permissions for this object and contexts.	If sender has the proper access and permissions, go to Step 9.
8	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Access and Permissions. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the sender lacks permission to subscribe to this object. Place the name of the object in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 12.
9	Check the Subscribers database to see if the caller is already subscribed to this object for the specified contexts.	If the caller is already subscribed to this object, go to Step 2.
10	Add a record in the Subscribers database to indicate that <code>SIF_SourceId</code> is a subscriber of this object's <code>SIF_Events</code> in the specified contexts. If an error occurs, add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> .	If an error occurs go to Step 12; otherwise go to Step 2.
11	Add a <code>SIF_Status</code> element indicating success to the <code>SIF_Ack</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
12	Undo all changes to the Subscribers database. Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.

Table 4.2.2.6-1: *SIF_Subscribe Handling*

4.2.2.7 SIF_Unsubscribe

If an agent wishes to cancel one or more subscriptions, the `SIF_Unsubscribe` message is used. Events already queued for delivery prior to unsubscription will be delivered.

The `SIF_Unsubscribe` message can contain subscription requests for multiple objects. The ZIS must treat all of the objects as a set, if there is an error with one of the objects then there should be no change to the Subscribers database.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> .	Go to Step 3.
2	Examine the message to determine whether any more unsubscriptions need to be processed.	Go to Step 6 if there are no further objects to process in the message.
3	Retrieve the name of the next object. If not otherwise performed in initial message validation, check whether the object name is valid (e.g. valid/supported object with events reported).	If the object name is valid, go to Step 5.

Step	Process	Flow Control
4	Add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> . Set <code>SIF_Error/SIF_Category</code> to indicate Subscription and set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the object is invalid. Place the name of the invalid object in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 9.
5	If no <code>SIF_Context</code> is specified, the context is <code>SIF_Default</code> . Otherwise check that each <code>SIF_Context</code> supplied in <code>SIF_Contexts</code> is supported.	If they are all supported, go to Step 7.
6	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Generic Message Handling. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate a context is not supported. Place the name of the context in <code>SIF_Error/SIF_ExtendedDesc</code> .	Go to Step 12.
7	If it exists, remove the record in the Subscribers database that marks <code>SIF_SourceId</code> as a subscriber of this object's <code>SIF_Events</code> in the specified contexts. If an error occurs, add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> .	If an error occurs go to Step 9, otherwise go to Step 2.
8	Add a <code>SIF_Status</code> element indicating success to the <code>SIF_Ack</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
9	Undo all changes to the Subscribers database. Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.

Table 4.2.2.7-1: *SIF_Unsubscribe Handling*

4.2.2.8 SIF_Provision

An Agent is registering its support for various messages with regard to various objects. Settings supplied replace any previously recorded settings for the Agent.

Step	Process	Flow Control
1	Prepare <code>SIF_Ack</code> .	
2	Process <code>SIF_ProvideObjects</code> as provide.	On error go to step 13.
3	Process objects not in <code>SIF_ProvideObjects</code> as unprovide.	On error go to step 13.
4	Process <code>SIF_SubscribeObjects</code> as subscribe.	On error go to step 13.
5	Process objects not in <code>SIF_SubscribeObjects</code> as unsubscribe.	On error go to step 13.
6	Process <code>SIF_PublishAddObjects</code> .	On error go to step 13.
7	Process <code>SIF_PublishChangeObjects</code> .	On error go to step 13.
8	Process <code>SIF_PublishDeleteObjects</code> .	On error go to step 13.
9	Process <code>SIF_RequestObjects</code> .	On error go to step 13.
10	Process <code>SIF_RespondObjects</code> .	On error go to step 13.
11	Save changes.	
12	Return success <code>SIF_Ack</code> .	Stop processing.
13	Roll back any changes.	
14	Return error <code>SIF_Ack</code> .	Stop processing.

Table 4.2.2.8-1: *SIF_Provision Handling*

4.2.2.9 SIF_Event

When an application has made a change in an object that is part of the Zone and for which the application has declared the ability to generate `SIF_Events`, the agent will send a `SIF_Event` message to its Zone Integration Server so the framework may distribute it.

Step	Process	Flow Control
1	Examine message and retrieve the name of the object. Check whether the object name is valid (e.g. valid/supported object with events reported).	If object name is valid, go to Step 3.

Step	Process	Flow Control
2	Add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> . Set <code>SIF_Error/SIF_Category</code> to indicate Event Reporting and set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the event is invalid. Place the name of the invalid object in <code>SIF_Error/SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
3	If no <code>SIF_Context</code> is specified, the context is <code>SIF_Default</code> . Otherwise check that each <code>SIF_Context</code> supplied in <code>SIF_Contexts</code> is supported.	If they are all supported, go to Step 5.
4	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Generic Message Handling. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate a context is not supported. Place the name of the context in <code>SIF_Error/SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
5	Using the <code>SIF_SourceId</code> , consult the ACL to determine if the sender has the proper access and permissions for this object in the specified contexts.	If sender has the proper access and permissions, go to Step 7.
6	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Access and Permissions. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the sender lacks permission to publish events pertaining to this object (use general <code>SIF_Event</code> error code, or specific Add, Change, Delete codes). Place the name of the object in <code>SIF_Error/SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
7	Check the Subscriber database to see if there are any subscribers in the specified contexts for the <code>SIF_Event</code> .	Go to Step 9 if there are no subscribers for this object.
8	For each subscriber make a copy of the <code>SIF_Event</code> . If the ZIS supports XML filtering pass the copy to the XML filter logic. If an XML filter matched the root <code>SIF_Message</code> do not put the copy into the subscriber's queue and continue to the next subscriber. With the copy if more than one context is specified for the event, only one copy of the event is placed in the subscribing agent's queue. If the event cannot be placed into an individual agent's queue due to the agent's maximum buffer size or because the subscribing agent does not support the message version of the <code>SIF_Event</code> , it is RECOMMENDED that the ZIS log the inability to deliver the event. In addition, the ZIS MUST report a <code>SIF_LogEntry</code> event with the appropriate error category and code, containing a copy of the <code>SIF_Header</code> from the original message. <code>SIF_LogEntry/SIF_Desc</code> must contain the <code>SourceId</code> of the agent that has failed to receive the message.	
9	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Status</code> element indicating success. Return a <code>SIF_Ack</code> to the caller.	Stop processing the message.

Table 4.2.2.9-1: *SIF_Event Handling*

4.2.2.10 SIF_Request

When an agent needs information from a Zone context it sends a `SIF_Request` message to the ZIS. If the `SIF_Request`'s header does not contain a `SIF_DestinationId` element, the ZIS will route the message to the Provider of the object referenced in the `SIF_Request`. If the header contains a `SIF_DestinationId`, the ZIS will route the message to the application referenced in the `SIF_DestinationId` if the security policies of the zone permit such routing. The ZIS will return a `SIF_Ack` message to the requesting agent to indicate whether or not it was able to process the `SIF_Request` message.

After the ZIS returns a success `SIF_Ack` to the requester, the ZIS will route the `SIF_Request` to the responder and the requesting agent may expect to receive one or more `SIF_Response` messages sent by the responder. However, the responder may not be currently on-line or it may not be able to immediately satisfy the `SIF_Request`. Therefore, requesting agents must not depend upon a timely response to their `SIF_Request`.

If the ZIS returns an error `SIF_Ack`, the requesting agent will not receive any `SIF_Response` messages from a responder.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> .	
2	Retrieve the name of the object from the <code>ObjectName</code> attribute of <code>SIF_Query/SIF_QueryObject</code> and check whether it's a valid/supported object.	Go to Step 4 if the object name is valid.

Step	Process	Flow Control
3	Add a <code>SIF_Error</code> element to the <code>SIF_Ack</code> . Set <code>SIF_Error/SIF_Category</code> to indicate Request and Response and set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the object name is invalid. Place the name of the invalid object in <code>SIF_Error/SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
4	If no <code>SIF_Context</code> is specified, the context is <code>SIF_Default</code> . Otherwise check that the context supplied in <code>SIF_Contexts</code> is supported. If more than one context is specified, go to Step 5.	If the context is supported, go to Step 6.
5	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Generic Message Handling. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate a specified context is not supported or that multiple contexts are not supported, depending on the error. Place the name of the context in <code>SIF_Error/SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
6	Using the <code>SIF_SourceId</code> , consult the ACL to determine if the sender has the proper access and permissions for this object in the applicable context.	If sender has the proper access and permissions, go to Step 8.
7	Prepare a <code>SIF_Ack</code> containing a <code>SIF_Error</code> element. Set <code>SIF_Error/SIF_Category</code> to indicate Access and Permissions. Set <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the sender lacks permission to request this object. Place the name of the object in <code>SIF_Error/SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
8	Examine the <code>SIF_Request</code> header looking for a <code>SIF_DestinationId</code>	Go to Step 11 if a <code>SIF_DestinationId</code> was located.
9	No <code>SIF_DestinationId</code> was found. Examine the Providers database to locate the responder for the requested object in the applicable context.	Go to Step 12 if a Provider was located.
10	Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that no provider was found. Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
11	A <code>SIF_DestinationId</code> was specified indicating the responder. Confirm that the agent specified in <code>SIF_DestinationId</code> has permission to send <code>SIF_Response</code> messages for the requested data object in the applicable context.	Go to Step 10 if the agent does not have the necessary permission.
12	If it can be determined from ACL settings or settings recorded by <code>SIF_Provision</code> and/or <code>SIF_Provide</code> that the Responder cannot handle a <code>SIF_Query</code> for a given object or <code>SIF_ExtendedQuery</code> for any referenced object, or that the Responder doesn't handle extended queries in general, add a <code>SIF_Error</code> element with the applicable <code>SIF_Error/SIF_Category</code> and <code>SIF_Error/SIF_Code</code> (object not supported, query not supported, or <code>SIF_ExtendedQuery</code> not supported). Place an appropriate error message in <code>SIF_Desc</code> and/or <code>SIF_ExtendedDesc</code> . Return the <code>SIF_Ack</code> to the caller.	Stop processing the message.
13	If the ZIS supports SIF XML filter pass apply the SIF XML filter logic to the <code>SIF_Request</code> . If a rule applied to the root element <code>SIF_Message</code> then Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate that the <code>SIF_Request</code> was canceled due to a SIF XML filter rule. Return the <code>SIF_Ack</code> to the caller.	Stop processing the message if an error <code>SIF_Ack</code> was returned.
14	Deposit the <code>SIF_Request</code> in the responder's queue. If the request cannot be placed into an individual agent's queue due to the agent's maximum buffer size or because the destination agent does not support the message version of the <code>SIF_Request</code> , it is RECOMMENDED that the ZIS log the inability to deliver the request. In addition, the ZIS MUST report a <code>SIF_LogEntry</code> event with the appropriate error category and code, containing a copy of the <code>SIF_Header</code> from this message. <code>SIF_LogEntry/SIF_Desc</code> MUST contain the <code>SourceId</code> of the agent that has failed to receive the request.	
15	Return a <code>SIF_Ack</code> , with <code>SIF_Status</code> set to 0, to the caller to indicate that <code>SIF_Request</code> has been sent.	Stop processing the message.

Table 4.2.2.10-1: *SIF_Request Handling*

4.2.2.11 SIF_Response

When receiving a SIF_Response packet from an agent responding to a SIF_Request, the ZIS **MUST** perform the validation protocol below.

Step	Process	Flow Control
1	Prepare a SIF_Ack.	
2	Using the supplied SIF_RequestMsgId, look up the SIF_Request that initiated this response.	Go to Step 4 if the SIF_Request is found.
3	Add a SIF_Error element to the SIF_Ack. Set SIF_Error/SIF_Category to indicate Request and Response and set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the SIF_RequestMsgId is invalid. Place SIF_RequestMsgId in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	Stop processing the message.
4	Examine the SIF_MaxBufferSize specified in the SIF_Request message and compare it to the size of the SIF_Response packet.	If the SIF_Response packet is smaller than or equal to the SIF_MaxBufferSize specified in the original request, go to Step 6.
5	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Request and Response. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the SIF_MaxBufferSize is incorrect. Place a description of the SIF_MaxBufferSize and the actual size of the message received in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	Go to step 14.
6	Examine the SIF_DestinationId specified in the SIF_Response and compare it to the SIF_SourceId of the original request.	If the SIF_DestinationId is correct, go to Step 8.
7	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Request and Response. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the SIF_DestinationId is incorrect. Place a description of the SIF_DestinationId specified and the SIF_DestinationId expected in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	Go to step 14.
8	Examine the SIF_PacketNumber specified in the SIF_Response. If this is the first SIF_Response packet received, the SIF_PacketNumber must be set to a value of 1. Subsequent packets must be received in order with the SIF_PacketNumber set to 1 + the previous SIF_PacketNumber.	If the SIF_PacketNumber is correct, go to Step 10.
9	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Request and Response. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the SIF_PacketNumber is incorrect. Place a description of the SIF_PacketNumber specified and the SIF_PacketNumber expected in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	Go to step 14.
10	Examine the SIF_Version specified in the SIF_Response and compare it to the SIF_Versions allowed in the original request.	If the SIF version matches one of the SIF Versions requested in the SIF_Request, go to Step 12.
11	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Request and Response. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the SIF_Version is incorrect. Place a description of the version of the SIF_Response and versions allowed by the SIF_Request in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	Go to step 14.
12	If the ZIS supports SIF XML filter apply the xml filter rules to the SIF_Response.	
13	Place the SIF_Response packet in the requesting agent's queue.	Message processing is complete. Stop processing the message.

Step	Process	Flow Control
14	<p>Prepare a <code>SIF_Response</code> message with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.</p> <p>Add a <code>SIF_Error</code> element with the <code>SIF_Error/SIF_Category</code> set to indicate Request and Response and <code>SIF_Error/SIF_Code</code> and <code>SIF_Error/SIF_Desc</code> to indicate the reason that the <code>SIF_Response</code> packet was rejected.</p> <p>Add <code>SIF_PacketNumber</code> with a value set to set to 1 + the previous <code>SIF_PacketNumber</code> and <code>SIF_MorePackets</code> to No.</p> <p>Send the <code>SIF_Response</code> to the original requester. In addition, the ZIS MUST report a <code>SIF_LogEntry</code> event with the appropriate error category and code, containing a copy of the <code>SIF_Header</code> from the request. <code>SIF_LogEntry/SIF_ExtendedDesc</code> should contain information about why the message failed <code>SIF_Response</code> validation.</p>	<p>Stop processing the message.</p> <p>The ZIS must also guarantee that no additional <code>SIF_Response</code> packets for this <code>SIF_Request</code> will be accepted. Depending on the implementation, the ZIS may need to alter the <code>SIF_Request</code> cache it maintains to signal that the <code>SIF_Request</code> is no longer valid.</p> <p>The ZIS may remove the <code>SIF_Request</code> from the cache as the stream is closed.</p>

Table 4.2.2.11-1: *SIF_Response Handling*

4.2.2.12 SIF_Ping

An Agent is pinging your ZIS to see if it is reachable, "awake" and/or processing messages.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. If your ZIS is "awake," include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 0 (success). Otherwise you may optionally notify the Agent that your ZIS is asleep by returning a <code>SIF_Code</code> of 8 (receiver is sleeping).	
2	Return the <code>SIF_Ack</code> to the Agent.	Message processing complete (success).

Table 4.2.2.12-1: *SIF_Ping Handling*

4.2.2.13 SIF_Sleep

The Agent wants its state changed to "asleep." Upon successful state change, your ZIS **SHOULD** avoid sending messages to a Push-Mode Agent until receipt of a `SIF_Wakeup` message or that Agent re-registers, or be prepared to handle transport errors or the aforementioned acknowledgement. Whether the Agent is registered in Push or Pull mode, this state is communicated to other Agents in `SIF_ZoneStatus` and **MUST** be persisted accordingly. In addition to sending a `SIF_Wakeup` or `SIF_Register`, a Pull-Mode Agent can also change its state to "awake" by sending a `SIF_GetMessage`.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 0 (success). Change the state of the Agent to "asleep."	
2	Return the <code>SIF_Ack</code> to the Agent.	Message processing complete (success).

Table 4.2.2.13-1: *SIF_Sleep Handling*

4.2.2.14 SIF_Wakeup

An Agent wants its state changed to "awake," notifying the ZIS and other Agents of the state change. A ZIS **MUST** persist this state in order to communicate it to other Agents via `SIF_ZoneStatus`. When a Push-Mode Agent changes its state to "awake," the ZIS may also resume delivery of queued messages to the Agent.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 0 (success). Change the Agent's state to "awake."	
2	Return the <code>SIF_Ack</code> to the Agent.	Message processing complete (success).

Table 4.2.2.14-1: *SIF_Wakeup Handling*

4.2.2.15 SIF_GetZoneStatus

An Agent is requesting the status of the zone.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 0 (success). Reflect the current state of the zone in <code>SIF_Status/SIF_Data/SIF_ZoneStatus</code> .	
2	Return the <code>SIF_Ack</code> to the Agent.	Message processing complete (success).

Table 4.2.2.15-1: *SIF_GetZoneStatus Handling*

4.2.2.16 SIF_GetAgentACL

An Agent is requesting its access control permissions.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 0 (success). Communicate the Agent's ACL permissions in <code>SIF_Status/SIF_Data/SIF_AgentACL</code> .	
2	Return the <code>SIF_Ack</code> to the Agent.	Message processing complete (success).

Table 4.2.2.16-1: *SIF_GetZoneStatus Handling*

4.2.2.17 SIF_CancelRequests

If an Agent abandons or restarts a data collection using `SIF_RequestS`, whether or not the response stream has started, it is **RECOMMENDED** that it send one or more `SIF_CancelRequests` messages to the ZIS. Upon receipt of the `SIF_CancelRequests` message, the ZIS deletes corresponding `SIF_Request` messages from Agent queues and deletes its own state/tracking information regarding each request. Doing the latter ensures that if a Responder is still processing a request, the ZIS effectively ends the response stream upon receipt of the next `SIF_Response` packet by returning a `SIF_Error` with a `SIF_Category` of 8 (Request and

Response Error) and a `SIF_Code` of 10 (invalid `SIF_RequestMsgId` specified in `SIF_Response`). No changes to responding Agent behaviors are required as all agents in the SIF 2.x lifecycle have the capability to handle this error state.

When cancelling `SIF_Requests`, the ZIS also has the ability to send a `SIF_CancelRequests` message to Push-Mode Agents. Pull-Mode Responders cannot receive these messages, but any pending response handling is cancelled per the ZIS behavior above. When dealing with Push-Mode Agents, ZIS implementations must bear in mind that support for this message is optional for Push-Mode Agents.

When a cancelling Agent specifies a `NotificationType` of `Standard`, it is the responsibility of the ZIS to end the response stream to the requesting Agent by sending a `SIF_Response` packet with a `SIF_MorePackets` of `No` on the Responder's behalf.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 0 (success). Return the <code>SIF_Ack</code> to the Agent. (There are no error return values that apply to this message.)	Go to Step 2.
2	For each <code>SIF_RequestMsgId</code> element, perform the following steps.	If all <code>SIF_RequestMsgId</code> elements have been processed, processing is complete.
3	Using the supplied <code>SIF_RequestMsgId</code> , look up the <code>SIF_Request</code> that initiated this response.	Go to Step 2 if the <code>SIF_Request</code> is not found, or has already been completed with a "final" <code>SIF_Response</code> packet (<code>SIF_MorePackets</code> = No).
4	Examine the <code>SIF_SourceId</code> specified in the <code>SIF_Request</code> message and compare it to the <code>SIF_SourceId</code> in the <code>SIF_SystemControl</code> message.	If the <code>SIF_SourceId</code> is not the same, go to Step 2.
5	Close out the <code>SIF_Request</code> tracking state for the request so that no further tracking is performed.	
6	If the responding Agent has already received the request and is running in Push mode, send a <code>SIF_CancelRequests</code> message to that Agent. (Note: This could also be accomplished by packaging up all <code>SIF_RequestMsgIds</code> that apply to the same responding Agent and sending a single <code>SIF_CancelRequests</code> message.)	
7	Examine the value of <code>SIF_NotificationType</code> .	If set to <code>Standard</code> , go to Step 8. If set to <code>None</code> , go to Step 10.
8	Prepare a <code>SIF_Response</code> message with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_RequestMsgId</code> set to <code>SIF_MsgId</code> from the <code>SIF_Request</code> message.	
9	Add a <code>SIF_Error</code> element with the <code>SIF_Category</code> set to indicate Request and Response, with <code>SIF_Code</code> and <code>SIF_Desc</code> indicating 18 (SIF_Request cancelled by requesting agent). Add <code>SIF_PacketNumber</code> with a value set to the previous <code>SIF_PacketNumber</code> + 1. Set <code>SIF_MorePackets</code> to No. Place the <code>SIF_Response</code> in the requester's queue.	
10	Determine if there are any more <code>SIF_RequestMsgId</code> elements left to process.	Go to Step 2 if there are more <code>SIF_RequestMsgId</code> elements, otherwise processing is complete.

Table 4.2.2.17-1: `SIF_CancelRequests` Handling

4.2.2.18 `SIF_CancelServiceInputs`

If an Agent abandons or restarts a data collection using `SIF_ServiceInputs`, whether or not the response stream has started, it is **RECOMMENDED** that it send one or more `SIF_CancelServiceInputs` messages to the ZIS. Upon receipt of the `SIF_CancelServiceInputs` message, the ZIS deletes corresponding `SIF_ServiceInput` messages from Agent queues and deletes its own state/tracking information regarding each request. Doing the latter ensures that if a Responder is still processing a service input, the ZIS effectively ends the response stream upon receipt of the next `SIF_ServiceOutput` packet by returning a `SIF_Error`

with a `SIF_Category` of 14 (SIF Zone Service Error) and a `SIF_Code` of 8 (invalid `SIF_ServiceMsgId` specified in `SIF_ServiceOutput`). No changes to responding Agent behaviors are required as all agents in the SIF 2.x lifecycle have the capability to handle this error state.

When cancelling `SIF_ServiceInputs`, the ZIS also has the ability to send a `SIF_CancelServiceInputs` message to Push-Mode Agents. Pull-Mode Responders cannot receive these messages, but any pending response handling is cancelled per the ZIS behavior above. When dealing with Push-Mode Agents, ZIS implementations must bear in mind that support for this message is optional for Push-Mode Agents.

When a cancelling Agent specifies a `NotificationType` of `Standard`, it is the responsibility of the ZIS to end the response stream to the requesting Agent by sending a `SIF_ServiceOutput` packet with a `SIF_MorePackets` of `No` on the Responder's behalf.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply. Place the incoming <code>SIF_Header/SIF_SourceId</code> and <code>SIF_Header/SIF_MsgId</code> in <code>SIF_OriginalSourceId</code> and <code>SIF_OriginalMsgId</code> , respectively. Include a <code>SIF_Status</code> element with a <code>SIF_Code</code> of 0 (success). Return the <code>SIF_Ack</code> to the Agent. (There are no error return values that apply to this message.)	Go to Step 2.
2	For each <code>SIF_ServiceMsgId</code> element, perform the following steps.	If all <code>SIF_ServiceMsgId</code> elements have been processed, processing is complete.
3	Using the supplied <code>SIF_ServiceMsgId</code> , look up the <code>SIF_ServiceInput</code> that initiated this response.	Go to Step 2 if the <code>SIF_ServiceInput</code> is not found, or has already been completed with a "final" <code>SIF_ServiceOutput</code> packet (<code>SIF_MorePackets = No</code>).
4	Examine the <code>SIF_SourceId</code> specified in the <code>SIF_ServiceInput</code> message and compare it to the <code>SIF_SourceId</code> in the <code>SIF_SystemControl</code> message.	If the <code>SIF_SourceId</code> is not the same, go to Step 2.
5	Close out the <code>SIF_ServiceInput</code> tracking state for the request so that no further tracking is performed.	
6	If the responding Agent has already received the request and is running in Push mode, send a <code>SIF_CancelServiceInputs</code> message to that Agent. (Note: This could also be accomplished by packaging up all <code>SIF_ServiceMsgIds</code> that apply to the same responding Agent and sending a single <code>SIF_CancelServiceInputs</code> message.)	
7	Examine the value of <code>SIF_NotificationType</code> .	If set to <code>Standard</code> , go to Step 8. If set to <code>None</code> , go to Step 10.
8	Prepare a <code>SIF_ServiceOutput</code> message with <code>SIF_DestinationId</code> set to <code>SIF_SourceId</code> and <code>SIF_ServiceMsgId</code> set to <code>SIF_ServiceMsgId</code> from the <code>SIF_ServiceInput</code> message.	
9	<p>Add a <code>SIF_Error</code> element with the <code>SIF_Category</code> set to indicate SIF Zone Service, with <code>SIF_Code</code> and <code>SIF_Desc</code> indicating 15 (SIF_ServiceInput cancelled by requesting agent).</p> <p>Add <code>SIF_PacketNumber</code> with a value set to the previous <code>SIF_PacketNumber</code> + 1. Set <code>SIF_MorePackets</code> to <code>No</code>.</p> <p>Place the <code>SIF_ServiceOutput</code> in the requester's queue.</p>	
10	Determine if there are any more <code>SIF_ServiceMsgId</code> elements left to process.	Go to Step 2 if there are more <code>SIF_ServiceMsgId</code> elements, otherwise processing is complete.

Table 4.2.2.18-1: *SIF_CancelRequests Handling*

4.2.2.19 SIF_GetMessage

A Pull-Mode Agent is requesting the next message in its queue.

Step	Process	Flow Control
1	Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId , your Zone Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply. Place the incoming SIF_Header/SIF_SourceId and SIF_Header/SIF_MsgId in SIF_OriginalSourceId and SIF_OriginalMsgId , respectively. If the Agent sending SIF_GetMessage is registered as a Pull-Mode Agent, go to step 3.	
2	The Agent is a Push-Mode Agent and is not allowed to send SIF_GetMessage . Include a SIF_Error/SIF_Category of 5 (Registration) and a SIF_Error/SIF_Code of 9 (Agent is registered in Push mode). Populate SIF_Desc and optionally SIF_ExtendedDesc as desired. Return the SIF_Ack to the Agent.	Message processing complete.
3	If the recorded state of the Pull-Mode Agent is "asleep," change that state to "awake." Is there a message available in the Agent's message queue, subject to Selective Message Blocking? If yes, go to step 5.	
4	There is no message currently available for the Agent. Include a SIF_Status/SIF_Code of 9 (no messages available). Return the SIF_Ack to the Agent.	Message processing complete.
5	The next available message in the Agent's queue, subject to Selective Message Blocking, can be delivered (it will be removed from the queue later per successful handling of a SIF_Ack from the Pull-Mode Agent). If SIF_Security is specified on the message and the connection from the Pull-Mode Agent does not meet the specified minimum encryption and/or authentication levels, or if the connection does not meet minimum encryption/authentication levels in the Zone, remove the message from the Agent's queue and return an appropriate SIF_Error . Otherwise include a SIF_Status/SIF_Code of 0 (success) and place the message in SIF_Status/SIF_Data .	Message processing complete.

Table 4.2.2.19-1: [SIF_GetMessage](#) Handling

4.2.2.20 [SIF_Ack](#) (from a Push-Mode Agent)

A Push-Mode Agent is sending a final [SIF_Ack](#) to end Selective Message Blocking (SMB).

Step	Process	Flow Control
1	Prepare a SIF_Ack message with SIF_Header containing a new GUID in SIF_MsgId , your Zone Id in SIF_SourceId and the current time in SIF_Timestamp ; other SIF_Header elements do not apply.	
2	Is SIF_Status/SIF_Code 3 (final SIF_Ack)?	If yes, go to Step 4.
3	<p>The Agent has violated protocol. End SMB if it has been invoked by the Agent and remove the blocked SIF_Event from the Agent's queue. Indicate SIF_Error/SIF_Category of 13 (SMB Error) and SIF_Error/SIF_Code 3 (final SIF_Ack expected).</p> <p>It is RECOMMENDED that your ZIS log the error. Your ZIS MAY post a SIF_LogEntry Add event with the same error category and code above, containing a copy of the SIF_Header of the message.</p>	Return the SIF_Ack to the Agent. Message handling complete (error).
4	Does SIF_OriginalMsgId match the SIF_MsgId for the SIF_Event that was blocked in SMB, if any?	If yes, go to Step 6.
5	<p>The Agent has violated protocol. As there can be only one event blocked by SMB, end SMB for the agent and remove the blocked SIF_Event from the Agent's queue, if any. Indicate SIF_Error/SIF_Category of 13 (SMB Error) and SIF_Error/SIF_Code 4 (incorrect SIF_MsgId in final SIF_Ack).</p> <p>It is RECOMMENDED that your ZIS log the error. Your ZIS MAY post a SIF_LogEntry Add event with the same error category and code above, containing a copy of the SIF_Header of the message.</p>	Return the SIF_Ack to the Agent. Message handling complete (error).
6	SMB has been ended by the Agent. Removed the blocked SIF_Event from the Agent's queue. Place 0 in SIF_Status/SIF_Code .	Return the SIF_Ack to the Agent. Message handling complete (success).

Table 4.2.2.20-1: [SIF_Ack](#) Handling

4.2.2.21 SIF_Ack (from a Pull-Mode Agent)

A Pull-Mode Agent is acknowledging a message it has retrieved using `SIF_GetMessage`. This typically leads to the message in question being removed from the Agent's queue. The Agent may also invoke Selective Message Blocking when acknowledging an event, blocking delivery of subsequent `SIF_Events` until Selective Message Blocking is ended by the Agent.

Step	Process	Flow Control
1	Prepare a <code>SIF_Ack</code> message with <code>SIF_Header</code> containing a new GUID in <code>SIF_MsgId</code> , your Zone Id in <code>SIF_SourceId</code> and the current time in <code>SIF_Timestamp</code> ; other <code>SIF_Header</code> elements do not apply.	
2	Is <code>SIF_Error</code> present?	If yes, go to Step 14.
3	Is <code>SIF_Status/SIF_Code</code> 1 (immediate <code>SIF_Ack</code>)?	If no, go to Step 5.
4	If no message matches <code>SIF_OriginalMsgId</code> , set <code>SIF_Error/SIF_Category</code> to 12 (Generic Message Handling) and <code>SIF_Error/SIF_Code</code> to 6 (no such message). Otherwise remove the identified message from the Agent's queue and set <code>SIF_Status/SIF_Code</code> to 0.	Return <code>SIF_Ack</code> . Message handling complete.
5	Is <code>SIF_Status/SIF_Code</code> 2 (intermediate <code>SIF_Ack</code>)?	If no, go to Step 7.
6	If no message matches <code>SIF_OriginalMsgId</code> , set <code>SIF_Error/SIF_Category</code> to 12 (Generic Message Handling) and <code>SIF_Error/SIF_Code</code> to 6 (no such message). If the message identified is not a <code>SIF_Event</code> , set <code>SIF_Category</code> to 13 (SMB Error) and <code>SIF_Code</code> to 2 (SMB can only be invoked on a <code>SIF_Event</code>). Otherwise invoke SMB on the identified <code>SIF_Event</code> , persisting <code>SIF_OriginalMsgId</code> , and set <code>SIF_Status/SIF_Code</code> to 0. This event is blocked and all <code>SIF_Events</code> are frozen.	Return <code>SIF_Ack</code> . Message handling complete.
7	Is <code>SIF_Status/SIF_Code</code> 3 (final <code>SIF_Ack</code>)?	If no, go to Step 9.
8	If SMB has not been invoked or the message identified by <code>SIF_OriginalMsgId</code> doesn't exist or doesn't match the <code>SIF_Event</code> blocked by SMB, set <code>SIF_Error/SIF_Category</code> to 13 (SMB Error) and <code>SIF_Error/SIF_Code</code> to 4 (incorrect <code>SIF_MsgId</code> in final <code>SIF_Ack</code>). (In the case of SMB having been invoked but the message not matching, end SMB, remove the message blocked by SMB from the Agent's queue and unfreeze delivery of events.) Otherwise end SMB, remove the identified message from the Agent's queue and unfreeze delivery of events. Set <code>SIF_Status/SIF_Code</code> to 0.	Return <code>SIF_Ack</code> . Message handling complete.
9	Is <code>SIF_Status/SIF_Code</code> 7 (already have this <code>SIF_MsgId</code> from you)?	If no, go to Step 11.
10	The ZIS cannot correct this, as the <code>SIF_MsgId</code> originates from an Agent and can't be changed without other repercussions. Remove the message from the Agent's queue. Set <code>SIF_Status/SIF_Code</code> to 0.	Return <code>SIF_Ack</code> . Message handling complete.
11	Is <code>SIF_Status/SIF_Code</code> 8 (receiver is sleeping)?	If no, go to Step 13.
12	The Agent is stating it cannot process the message at this time; leave it as the next message to be delivered. Set <code>SIF_Status/SIF_Code</code> to 0.	Return <code>SIF_Ack</code> . Message handling complete.
13	The Agent has violated protocol. Indicate <code>SIF_Error/SIF_Category</code> of 12 (Generic Message Handling Error) and <code>SIF_Error/SIF_Code</code> 5 (protocol error) for the message.	Return <code>SIF_Ack</code> . Message handling complete.
14	The Agent has indicated a <code>SIF_Error</code> condition. See Error Codes with <code>SIF_Category</code> and <code>SIF_Code</code> , and examine <code>SIF_Desc</code> and <code>SIF_ExtendedDesc</code> , if included. If <code>SIF_Category</code> does not indicate a transport error, remove the message from the Agent's queue. Otherwise it remains the next message to be delivered. Set <code>SIF_Status/SIF_Code</code> to 0.	Return <code>SIF_Ack</code> . Message handling complete.

Table 4.2.2.21-1: SIF_Ack Handling

4.2.2.22 SIF_ServiceNotify

The design of SIF Zone Services allows a service definition to be defined as containing notification events that can be sent to the zone. When an application wishes to notify the zone that something has happened, it does so by sending a SIF_ServiceNotify to the ZIS.

Step	Process	Flow Control
1	Using the SIF_SourceId, consult the ACL to determine if the sender has the proper access and permissions for this service and operation in the specified context.	If sender has the proper access and permissions, go to step 3.
2	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate SIF Zone Service. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate ACL permission denied. Return the SIF_Ack to the caller.	Stop processing message.
3	Using the SIF_SourceId, consult the SIF_ZoneStatus to determine if the sender is a service provider.	If sender is a service provider go to step 5
4	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate Access and Permission Error. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the Agent has no permission to provide the service. Return the SIF_Ack to the caller.	Stop processing message.
5	Get SIF_ServiceMsgId value. Get SIF_PacketNumber value. Get SIF_MorePackets value. Use the SIF_ServiceMsgId to look up any previous state information in the packet tracking cache. Use the information retrieved and the information obtained from the SIF_Message/SIF_ServiceNotify to determine if the packet sequence for the SIF_PacketNumber is correct. The SIF_MorePackets is used to determine if this is the terminating packet.	If the SIF_PacketNumber is correct go to step 7
6	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate SIF Zone Service. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the SIF_PacketNumber is incorrect. Place a description of the SIF_PacketNumber specified and the SIF_PacketNumber expected in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	If the packet stream has not been terminated by the ZIS go to step 10.
7	Check the Subscriber database to see if there are any subscribers in the specified contexts for the specific SIF_Service and SIF_Operation from the SIF_ServiceNotify. The identification of a subscriber is based upon the service name and operation name.	Go to Step 9 if there are no subscribers for this notification event.

Step	Process	Flow Control
8	Place a copy of the SIF_ServiceNotify message into each subscribing agent's queue. If more than one context is specified for the event, only one copy of the event is placed in the subscribing agent's queue. If the SIF_ServiceNotify cannot be placed into an individual agent's queue due to the agent's maximum buffer size or because the subscribing agent does not support the message version of the SIF_ServiceNotify, it is RECOMMENDED that the ZIS log the inability to deliver the event. In addition, the ZIS MUST report a SIF_LogEntry event with the appropriate error category and code, containing a copy of the SIF_Header from the original message. SIF_LogEntry/SIF_Desc must contain the SourceId of the agent that has failed to receive the message.	
9	Prepare a SIF_Ack containing a SIF_Status element indicating success. Return a SIF_Ack to the caller.	Stop processing the message.
10	<p>Prepare a SIF_ServiceNotify message with a copy of the information in the original SIF_ServiceNotify except for the SIF_Body.</p> <p>Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate SIF Zone Service and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the reason that the SIF_ServiceNotify packet was rejected.</p> <p>Add SIF_PacketNumber with a value set to set to 1 + the previous SIF_PacketNumber and SIF_MorePackets to No.</p> <p>Send the SIF_ServiceNotify to the subscribers. In addition, the ZIS MUST report a SIF_LogEntry event with the appropriate error category and code, containing a copy of the SIF_Header from the request. SIF_LogEntry/SIF_ExtendedDesc should contain information about why the message failed SIF_ServiceNotify validation.</p>	<p>Stop processing the message.</p> <p>The ZIS must also guarantee that no additional SIF_ServiceNotify packets for this SIF_ServiceMsgId will be accepted. Depending on the implementation, the ZIS may need to alter the SIF_ServiceNotify cache it maintains to signal that the SIF_ServiceNotify/SIF_ServiceMsgId is no longer valid.</p> <p>The ZIS may remove the SIF_ServiceNotify/SIF_ServiceMsgId from the cache as the stream is closed.</p>

Table 4.2.2.22-1: SIF_ServiceInput Handling

4.2.2.23 SIF_ServiceInput

When an agent wishes to invoke an operation on a SIF Zone Service, it sends a SIF_ServiceInput message to the ZIS. If the SIF_ServiceInput's header does not contain a SIF_DestinationId element, the ZIS will route the message to the Provider of the service referenced in the SIF_ServiceInput. If the header contains a SIF_DestinationId, the ZIS will route the message to the application referenced in the SIF_DestinationId if the security policies of the zone permit such routing. The ZIS will return a SIF_Ack message to the requesting agent to indicate whether or not it was able to process the SIF_ServiceInput message.

After the ZIS returns a success SIF_Ack to the requester, the ZIS will route the SIF_ServiceInput to the responder and the requesting agent may expect to receive one or more SIF_ServiceOutput messages sent by the responder. However, the responder may not be currently on-line or it may not be able to immediately satisfy the SIF_ServiceInput. Therefore, requesting agents must not depend upon a timely response to their SIF_ServiceInput.

If the ZIS returns an error SIF_Ack, the requesting agent will not receive any SIF_ServiceOutput messages from a responder.

Step	Process	Flow Control
1	Using the SIF_SourceId, consult the ACL to determine if the sender has the proper access and permissions for this service and operation in the specified context.	If sender has the proper access and permissions, go to step 3.

Step	Process	Flow Control
2	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate SIF Zone Service. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate ACL permission denied. Return the SIF_Ack to the caller.	Stop processing message.
3	Get SIF_ServiceMsgId value. Get SIF_PacketNumber value. Get SIF_MorePackets value. Use the SIF_ServiceMsgId to look up any previous state information in the packet tracking cache. Use the information retrieved and the information obtained from the SIF_Message/SIF_ServiceInput to determine if the packet sequence for the SIF_PacketNumber is correct. The SIF_MorePackets is used to determine if this is the terminating packet.	If the SIF_PacketNumber is correct go to step 5
4	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate SIF Zone Service. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the SIF_PacketNumber is invalid. Place a description of the SIF_PacketNumber specified and the SIF_PacketNumber expected in SIF_Error/SIF_ExtendedDesc. Return the SIF_Ack to the caller.	If the packet stream has not been terminated by the ZIS go to step 13.
5	Examine the SIF_ServiceInput header looking for a SIF_DestinationId	Go to Step 8 if a SIF_DestinationId was located.
6	No SIF_DestinationId was found. Examine the Providers database to locate the responder for the requested service in the applicable context.	Go to Step 8 if a Provider was located.
7	Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate SIF Zone Service and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that no provider was found. Return the SIF_Ack to the caller.	Stop processing the message.
8	Examine the SIF_ServiceInput header looking for a SIF_Version. If the SIF_ServiceOutput does not match any SIF_Version from SIF_ServiceInput: Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate SIF Zone Service and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that a version mismatch occurred. Return the SIF_Ack to the caller	Go to Step 9 if a version match was made. If a version match is not made, complete the step then stop processing the message.
9	A SIF_DestinationId was specified indicating the responder and version numbers match. Confirm that the agent specified in SIF_DestinationId has permission to send SIF_ServiceOutput messages for the requested data object in the applicable context.	Go to Step 11 if the agent has the necessary permission.
10	If it can be determined from ACL settings that the Responder cannot handle a SIF_ServiceInput for the given service operation, add a SIF_Error element with the applicable SIF_Error/SIF_Category and SIF_Error/SIF_Code. Place an "ACL permission denied" error message in SIF_Desc and/or SIF_ExtendedDesc. Return the SIF_Ack to the caller.	Stop processing the message.

Step	Process	Flow Control
11	Deposit the SIF_ServiceInput in the responder's queue. If the request cannot be placed into an individual agent's queue due to the agent's maximum buffer size or because the destination agent does not support the message version of the SIF_ServiceInput, it is RECOMMENDED that the ZIS log the inability to deliver the request. In addition, the ZIS MUST report a SIF_LogEntry event with the appropriate error category and code, containing a copy of the SIF_Header from this message. SIF_LogEntry/SIF_Desc MUST contain the SourceId of the agent that has failed to receive the service request.	If the message cannot be put into the agent's queue and the SIF_PacketNumber is greater than 1 go to step 13.
12	Prepare a SIF_Ack containing a SIF_Status element indicating success. Return a SIF_Ack to the caller.	Stop processing the message.
13	<p>Prepare a SIF_ServiceInput message with a copy of the information in the original SIF_ServiceInput except for the SIF_Body.</p> <p>Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate SIF Zone Service and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the reason that the SIF_ServiceInput packet was rejected.</p> <p>Add SIF_PacketNumber with a value set to set to 1 + the previous SIF_PacketNumber and SIF_MorePackets to NO.</p> <p>Send the SIF_ServiceInput to the target agent. In addition, the ZIS MUST report a SIF_LogEntry event with the appropriate error category and code, containing a copy of the SIF_Header from the request. SIF_LogEntry/SIF_ExtendedDesc should contain information about why the message failed SIF_ServiceInput validation.</p>	<p>Stop processing the message.</p> <p>The ZIS must also guarantee that no additional SIF_ServiceInput packets for this SIF_ServiceMsgId will be accepted. Depending on the implementation, the ZIS may need to alter the SIF_ServiceInput cache it maintains to signal that the SIF_ServiceInput/SIF_ServiceMsgId is no longer valid.</p> <p>The ZIS may remove the SIF_ServiceInput/SIF_ServiceMsgId from the cache as the stream is closed.</p>

Table 4.2.2.23-1: SIF_ServiceInput Handling

4.2.2.24 SIF_ServiceOutput

When receiving a SIF_ServiceOutput packet from an agent responding to a SIF_ServiceInput, the ZIS **MUST** perform the validation protocol below.

Step	Process	Flow Control
1	Using the SIF_SourceId, consult the ACL to determine if the sender has the proper access and permissions for this service and operation in the specified context.	If sender has the proper access and permissions, go to step 3.
2	Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate SIF Zone Service. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate ACL permission denied. Return the SIF_Ack to the caller.	Stop processing message.

Step	Process	Flow Control
3	<p>Get SIF_ServiceMsgId value. Get SIF_PacketNumber value. Get SIF_MorePackets value.</p> <p>Use the SIF_ServiceMsgId to look up any previous state information in the packet tracking cache. Use the information retrieved and the information obtained from the SIF_Message/SIF_ServiceInput to determine if the packet sequence for the SIF_PacketNumber is correct. The SIF_MorePackets is used to determine if this is the terminating packet. The cache should also indicate if there was a SIF_ServiceInput that initiated this SIF_ServiceOutput. Also verify the SIF_ServiceOutput does not exceed the SIF_MaxBufferSize specified in the SIF_ServiceInput. The SIF_Version of the SIF_ServiceOutput MUST be in the range of versions specified in the SIF_ServiceInput.</p>	If the validation is correct go to step 5
4	<p>Prepare a SIF_Ack containing a SIF_Error element. Set SIF_Error/SIF_Category to indicate SIF Zone Service. Set SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate "Packet number invalid, Buffer size exceeded."</p>	If the packet stream has not been terminated by the ZIS go to step 9.
5	<p>Examine the SIF_ServiceInput header looking for a SIF_DestinationId. Verify the SIF_DestinationId matches the SIF_SourceId in the original SIF_ServiceInput.</p>	Go to Step 7 if a SIF_DestinationId was located.
6	<p>Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate SIF Zone Service and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate that SIF_DestinationId does not match the SIF_SourceId from SIF_ServiceInput. Return the SIF_Ack to the caller.</p>	Stop processing the message.
7	<p>Deposit the SIF_ServiceOutput in the Agent's queue. If the message cannot be placed into an individual agent's queue due to the agent's maximum buffer size or because the destination agent does not support the message version of the SIF_ServiceOutput, it is RECOMMENDED that the ZIS log the inability to deliver the message. In addition, the ZIS MUST report a SIF_LogEntry event with the appropriate error category and code, containing a copy of the SIF_Header from this message. SIF_LogEntry/SIF_Desc MUST contain the SourceId of the agent that has failed to receive the service request.</p>	If the message cannot be put into the agent's queue and the SIF_PacketNumber is greater than 1 go to step 9
8	<p>Prepare a SIF_Ack containing a SIF_Status element indicating success. Return a SIF_Ack to the caller.</p>	Stop processing the message.

Step	Process	Flow Control
9	<p>Prepare a SIF_ServiceOutput message with a copy of the information in the original SIF_ServiceOutput except for the SIF_Body.</p> <p>Add a SIF_Error element with the SIF_Error/SIF_Category set to indicate SIF Zone Service and SIF_Error/SIF_Code and SIF_Error/SIF_Desc to indicate the reason that the SIF_ServiceOutput packet was rejected.</p> <p>Add SIF_PacketNumber with a value set to set to 1 + the previous SIF_PacketNumber and SIF_MorePackets to NO.</p> <p>Send the SIF_ServiceOutput to the requesting agent. In addition, the ZIS MUST report a SIF_LogEntry event with the appropriate error category and code, containing a copy of the SIF_Header from the request. SIF_LogEntry/SIF_ExtendedDesc should contain information about why the message failed SIF_ServiceOutput validation.</p>	<p>Stop processing the message.</p> <p>The ZIS must also guarantee that no additional SIF_ServiceOutput packets for this SIF_ServiceMsgId will be accepted. Depending on the implementation, the ZIS may need to alter the SIF_ServiceOutput cache it maintains to signal that the SIF_ServiceOutput/SIF_ServiceMsgId is no longer valid.</p> <p>The ZIS may remove the SIF_ServiceOutput/SIF_ServiceMsgId from the cache as the stream is closed.</p>

Table 4.2.2.24-1: SIF_ServiceOutput Handling

5 Infrastructure

This section presents the XML structure for Infrastructure common elements, messages and objects in a tabular format for readers less versed in parsing formal XML schema definitions.

The Char(acteristics) column for all of the tables in this section use the following codes:

Code	Characteristic
M	Mandatory element or attribute
O	Optional element or attribute
C	Conditional element or attribute
MR	Mandatory and repeatable element
OR	Optional and repeatable element
CR	Conditional and repeatable element

Mandatory elements and attributes **MUST** be provided in the Infrastructure messages in which they appear. Infrastructure data objects (SIF_ZoneStatus, SIF_AgentACL) can be subject to SIF's request/response and event models; when impacted by these models (in a SIF_Event or in a SIF_Response), these objects follow the same conventions as listed in

Data Model.

5.1 Common Elements

5.1.1 SIF_ExtendedElements

This element is supported at the end of all SIF objects. The element is used to extend existing SIF objects with locally-defined elements. Extended elements **SHOULD NOT** be used to duplicate data that can be obtained from other SIF objects.

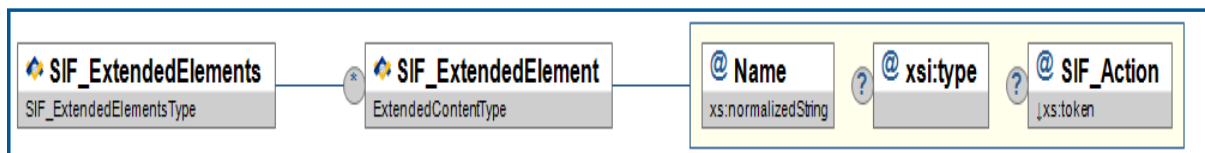


Figure 5.1.1-1: SIF_ExtendedElements

Element/@Attribute	Char	Description	Type
SIF_ExtendedElements		Allows an agent to include data not yet defined within a SIF data object as name/value pairs.	ActionList (SIF_ExtendedElement/@Name)
SIF_ExtendedElement	OR	A name/value pair, the name being contained in the Name attribute, the value being the element content.	ExtendedContentType

	Element/@Attribute	Char	Description	Type
@	Name	M	The name of the extended element. As it is possible that names for extended elements may collide from agent to agent, it is recommended that the names of extended elements be configurable in an agent, or that agents use URIs for the names of extended elements.	<code>xs:normalizedString</code>
@	<code>xs:type</code>	O	Allows type of element to be explicitly communicated.	
@	<code>SIF_Action</code>	O	In a <code>Change</code> event, this flag can be used to indicate an element has been deleted from the parent list container. At a minimum the key for the list must also be present.	values: <code>Delete</code>

Table 5.1.1-1: *SIF_ExtendedElements*

```

<SIF_ExtendedElements>
  <SIF_ExtendedElement Name="ApplicationSubmissionStatus">4</SIF_ExtendedElement>
  <SIF_ExtendedElement Name="DynamicXml">
    <Parent xmlns="http://myapplication.com">
      <Child n="1">one</Child>
      <Child n="2" />
      <Child n="3">three</Child>
    </Parent>
  </SIF_ExtendedElement>
  <SIF_ExtendedElement Name="Note">
    <xhtml:strong xmlns:xhtml="http://www.w3.org/1999/xhtml">Double</xhtml:strong>-check submission status.
  </SIF_ExtendedElement>
</SIF_ExtendedElements>

```

Example 5.1.1-1: *SIF_ExtendedElements*

5.1.2 SIF_Message

The `SIF_Message` element is the root element of all SIF messages.

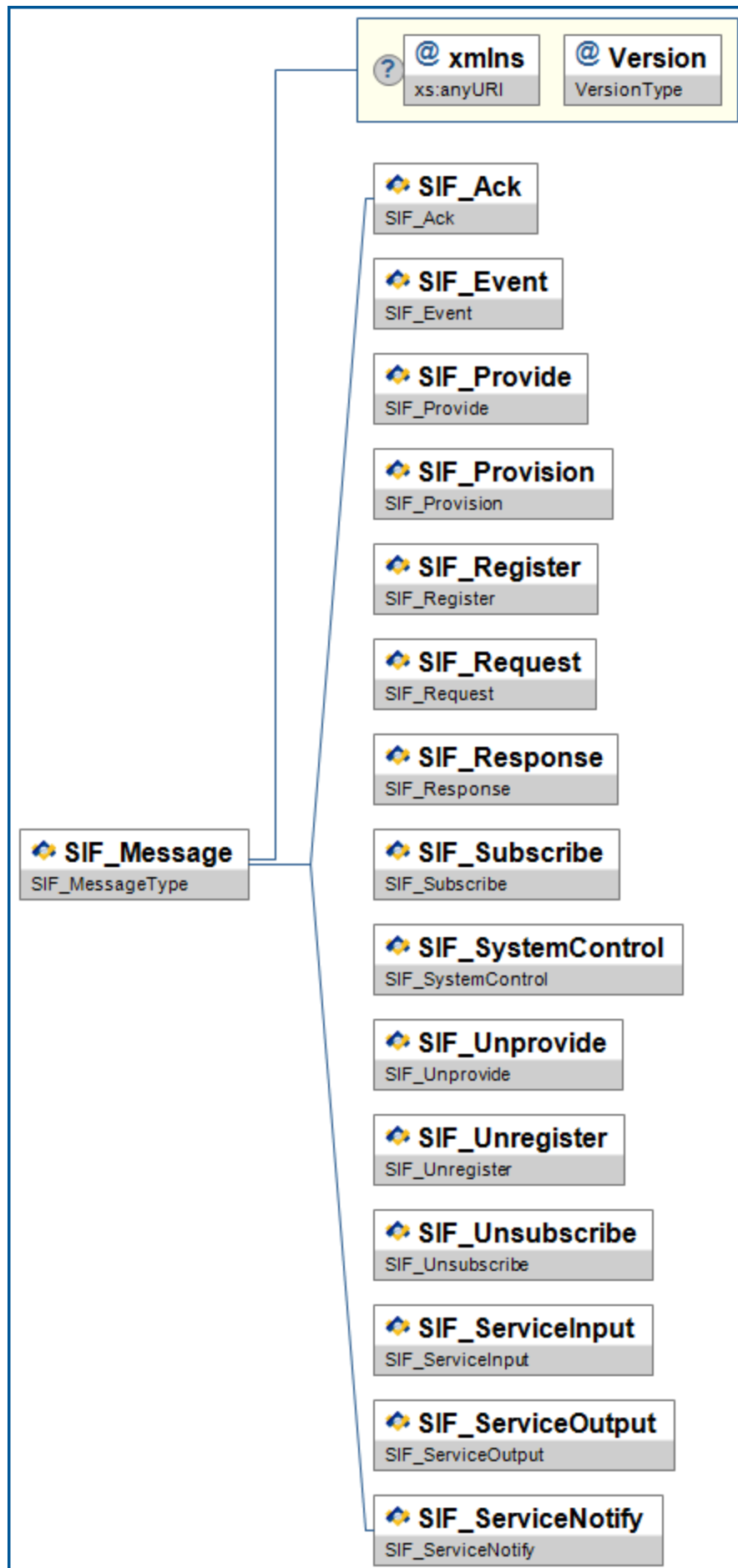


Figure 5.1.2-1: SIF_Message

	Element/@Attribute	Char	Description	Type
	SIF_Message		Contains one of the SIF message types.	choice of: SIF_Ack SIF_Event SIF_Provide SIF_Provision SIF_Register SIF_Request SIF_Response SIF_Subscribe SIF_SystemControl SIF_Unprovide SIF_Unregister SIF_Unsubscribe SIF_ServiceInput SIF_ServiceOutput SIF_ServiceNotify
@	xmlns	C	<p>The xmlns attribute specifies the XML namespace for SIF messages. For this version of the specification, the value of this attribute MUST be <code>http://www.sifinfo.org/infrastructure/2.x</code> . This XML namespace value will remain the same until the next major release of SIF (3.0).</p> <p>Note that one SIF_Message may be contained within another when a ZIS delivers a Pull-Mode Agent's next message in a SIF_Ack response to a SIF_GetMessage from the Pull-Mode Agent. If the default namespace specified for the child SIF_Message is the same as the default namespace of the parent SIF_Message, the xmlns attribute for the child message MAY be omitted.</p>	xs:anyURI
@	Version	M	The version of the SIF Implementation Specification that defines this message's XML structure. For this version of the specification, the value of this attribute is 2.5. This attribute can be used by ZIS and agent implementations to choose schema files to validate the message's XML.	VersionType

Table 5.1.2-1: SIF_Message

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  ...
</SIF_Message>
```

Example 5.1.2-1: SIF_Message

5.1.3 SIF_Header

SIF_Header is a common message header for all SIF messages.

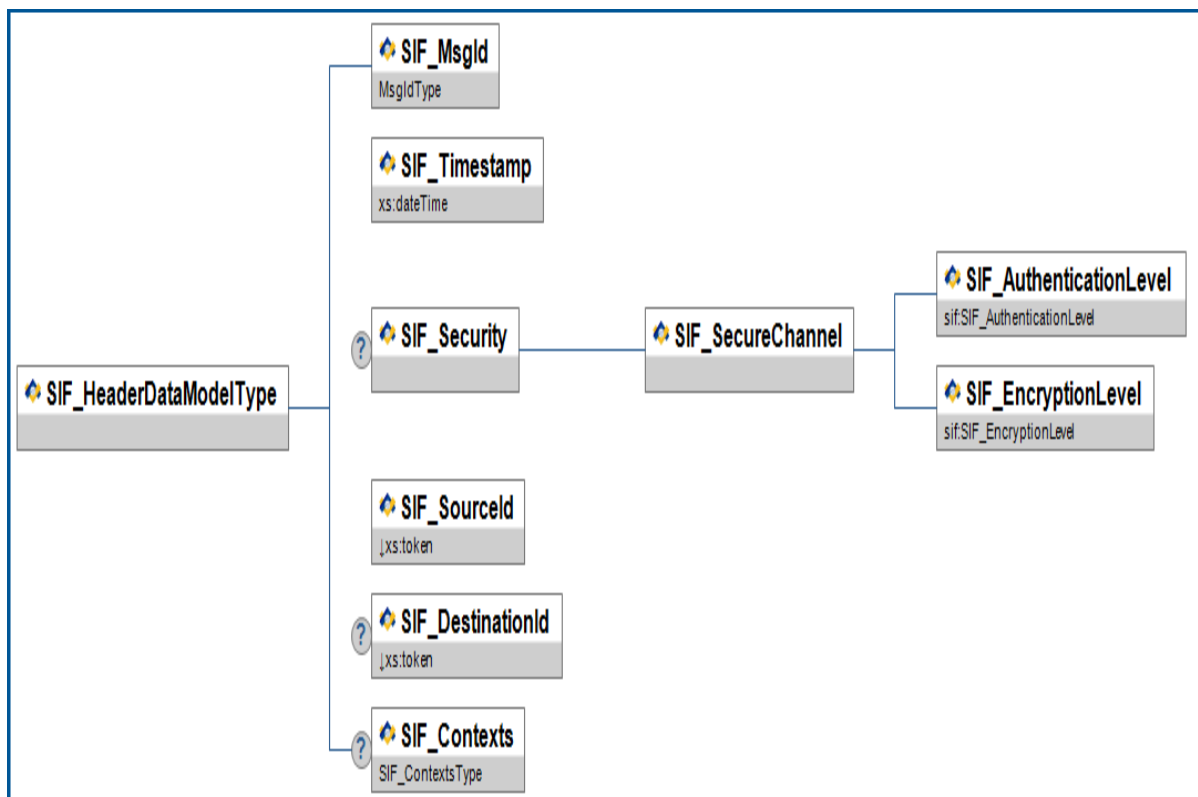


Figure 5.1.3-1: SIF_Header

Element/@ Attribute	Char	Description	Type
SIF_Header	M	Header information associated with a message.	
SIF_MsgId	M	SIF_MsgId is a globally unique message identifier from the Agent or ZIS that sends out the message.	MsgIdType
SIF_Timestamp	M	Timestamp of when the message was sent.	xs:dateTime
SIF_Security	O	This element allows an originating agent to specify security requirements that the ZIS must ensure upon delivery of the message to recipient agents. SIF_Security is only examined and processed by a ZIS on SIF_Request, SIF_Response, SIF_Event, SIF_ServiceNotify, SIF_ServiceInput, and SIF_ServiceOutput messages. In this version of the specification, SIF_Security is ignored on all other messages; its use on other messages is reserved for future versions of this specification.	
SIF_Security/SIF_SecureChannel	M	The originating agent uses this element to specify security requirements for the channel between the ZIS and any recipient agents at delivery time. The ZIS must ensure these requirements are met for this message when delivered to other agents.	

Element/@ Attribute	Char	Description	Type
SIF_Security/SIF_SecureChannel/ SIF_AuthenticationLevel	M	The minimum level of authentication required by the message originator to be considered a secure channel upon message delivery to other agents.	SIF_AuthenticationLevel
SIF_Security/SIF_SecureChannel/ SIF_EncryptionLevel	M	The minimum level of encryption required by the message originator to be considered a secure channel upon message delivery to other agents.	SIF_EncryptionLevel
SIF_SourceId	M	The SIF_SourceId is the Id of the originator of the message. Each source needs to have a zone unique case-sensitive identifier.	xs:token <div>xs:maxLength64</div>
SIF_DestinationId	C	<p>This element represents the Id of the recipient of the message and may be present as follows:</p> <p>SIF_Response messages MUST have this element set to the SIF_SourceId of the originator of the SIF_Request message. The ZIS will use this information to route the SIF_Response to the requesting agent.</p> <p>SIF_ServiceOutput messages MUST have this element set to the SIF_SourceId of the originator of the SIF_ServiceInput message. The ZIS will use this information to route the SIF_ServiceOutput to the requesting agent.</p> <p>SIF_Request messages MAY have this element set to the Id of a specific agent if the requesting agent wishes to direct the SIF_Request to a specific responder. If present, the ZIS will route the SIF_Request to the agent referenced in the element subject to the access control policies in effect for the zone.</p> <p>SIF_ServiceInput messages MAY have this element set to the Id of a specific agent if the invoking agent wishes to direct the SIF_ServiceInput to a specific responder. If present, the ZIS will route the SIF_ServiceInput to the agent referenced in this element; otherwise, if not present, the ZIS will route the message to the Provider of the service referenced in the SIF_ServiceInput. In both cases the ZIS will route messages subject to the access control policies in effect for the zone.</p> <p>SIF_Event and SIF_ServiceNotify messages MAY have this element set to the ID of a specific agent. If present, the ZIS will route the SIF_Event or SIF_ServiceNotify to the agent(s) referenced in the element, subject to the access control policies in effect for the zone.</p> <p>This element SHOULD NOT be used in any other SIF Infrastructure messages. If the element is present, it will be ignored by the ZIS.</p>	xs:token <div>xs:maxLength64</div>

Element/@Attribute	Char	Description	Type
SIF_Contexts	O	Contains each SIF Context that applies to the message. If omitted, the applicable context is SIF_Default. SIF_Context is repeatable for SIF_Events, not repeatable for SIF_Request or SIF_Response.	SIF_Contexts

Table 5.1.3-1: SIF_Header

```

<SIF_Header>
  <SIF_MsgId>A3E90785EFDA330DACB00785EFDA330D</SIF_MsgId>
  <SIF_Timestamp>2006-02-18T14:30:00-05:00</SIF_Timestamp>
  <SIF_SourceId>RamseySIS</SIF_SourceId>
</SIF_Header>

```

Example 5.1.3-1: SIF_Header

```

<SIF_Header>
  <SIF_MsgId>A3E90785EFDA330DACB00785EFDA330E</SIF_MsgId>
  <SIF_Timestamp>2006-03-11T08:39:49-08:00</SIF_Timestamp>
  <SIF_Security>
    <SIF_SecureChannel>
      <SIF_AuthenticationLevel>3</SIF_AuthenticationLevel>
      <SIF_EncryptionLevel>4</SIF_EncryptionLevel>
    </SIF_SecureChannel>
  </SIF_Security>
  <SIF_SourceId>RamseyLIB</SIF_SourceId>
  <SIF_DestinationId>RamseySIS</SIF_DestinationId>
</SIF_Header>

```

Example 5.1.3-2: SIF_Header

5.1.4 SIF_EncryptionLevel

The minimum level of encryption required by the message originator to be considered a secure channel upon message delivery to other agents.

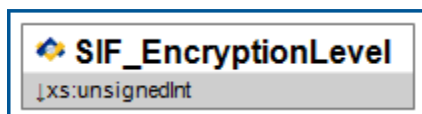


Figure 5.1.4-1: SIF_EncryptionLevel

Element/@Attribute	Char	Description	Type
SIF_EncryptionLevel		The minimum level of encryption required by the message originator to be considered a secure channel upon message delivery to other agents.	values: 0 No encryption required 1 Symmetric key length of at least 40 bits is to be used 2 Symmetric key length of at least 56 bits is to be used 3 Symmetric key length of at least 80 bits is to be used 4 Symmetric key length of at least 128 bits is to be used

Table 5.1.4-1: SIF_EncryptionLevel

5.1.5 SIF_AuthenticationLevel

The minimum level of authentication required by the message originator to be considered a secure channel upon message delivery to other agents.

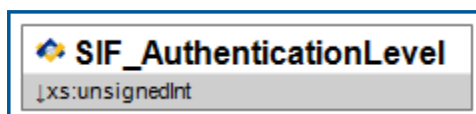


Figure 5.1.5-1: SIF_AuthenticationLevel

Element/@Attribute	Char	Description	Type
SIF_AuthenticationLevel		The minimum level of authentication required by the message originator to be considered a secure channel upon message delivery to other agents.	values: 0 No authentication required and a valid certificate does not need to be presented. 1 A valid certificate must be presented. 2 A valid certificate from a trusted certificate authority must be presented. 3 A valid certificate from a trusted certificate authority must be presented and the CN field of the certificate's Subject entry must match the host sending the certificate.

Table 5.1.5-1: SIF_AuthenticationLevel

5.1.6 SIF_Contexts

A list of SIF contexts that applies to a message or operation. Typically where used as an optional element, the omission of this element implies the SIF_Default context applies.



Figure 5.1.6-1: SIF_Contexts

Element/@Attribute	Char	Description	Type
SIF_Contexts		A list of SIF contexts that applies to a message or operation. Typically where used as an optional element, the omission of this element implies the SIF_Default context applies.	List
SIF_Context	MR		SIF_Context

Table 5.1.6-1: SIF_Contexts

5.1.7 SIF_Context

The name of a SIF Context that applies to a message or operation.

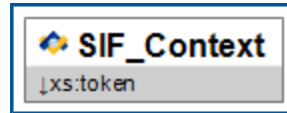


Figure 5.1.7-1: SIF_Context

Element/@Attribute	Char	Description	Type
SIF_Context		The name of a SIF Context that applies to a message or operation.	<code>xs:token</code> <code>xs:maxLength</code> 64

Table 5.1.7-1: SIF_Context

5.1.8 SIF_Protocol

Contains protocol information regarding a ZIS or Agent.

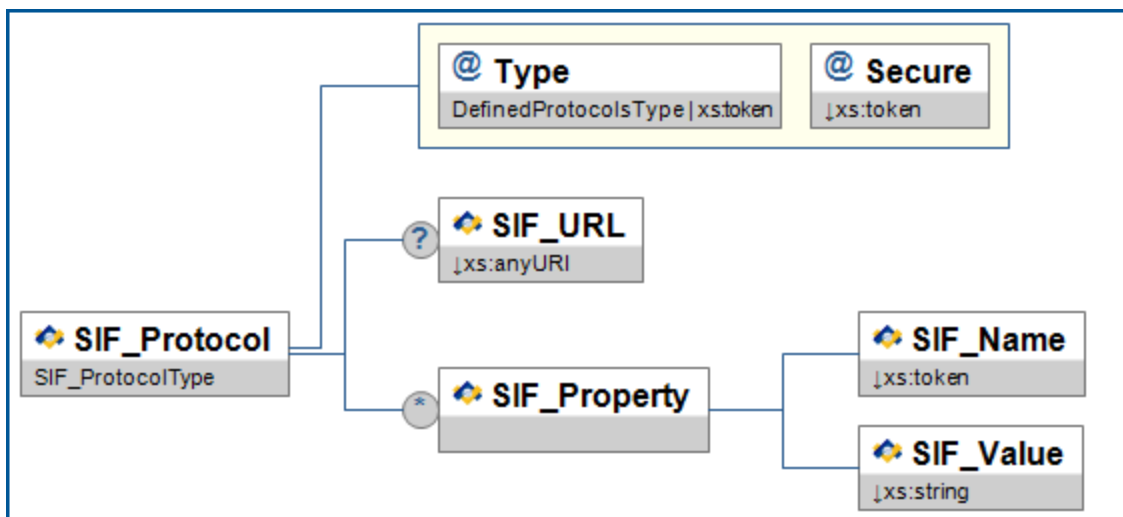


Figure 5.1.8-1: SIF_Protocol

Element/@Attribute	Char	Description	Type
SIF_Protocol	C	Contains protocol information regarding a ZIS or Agent.	
@ Type	M	The type of protocol to use (HTTPS, HTTP or an implementation-defined protocol).	union of: <code>DefinedProtocolsType</code> <code>xs:token</code>
@ Secure	M	Whether the protocol provides a secure channel.	values: Yes No
SIF_URL	C	This element is required if the protocol is HTTPS or HTTP. It contains the https or http URL for contacting the agent.	<code>xs:anyURI</code> <code>xs:maxLength</code> 256

Element/@Attribute	Char	Description	Type
SIF_Property	OR	May contain zero or more SIF_Property elements containing SIF_Name/SIF_Value pairs describing any protocol settings required to ensure proper communication.	
SIF_Property/SIF_Name	M	Property name.	xs:token xs:maxLength 64
SIF_Property/SIF_Value	M	Property value.	xs:string xs:maxLength 256

Table 5.1.8-1: SIF_Protocol

5.1.9 SIF_Status

This element is used to signal a successful response.

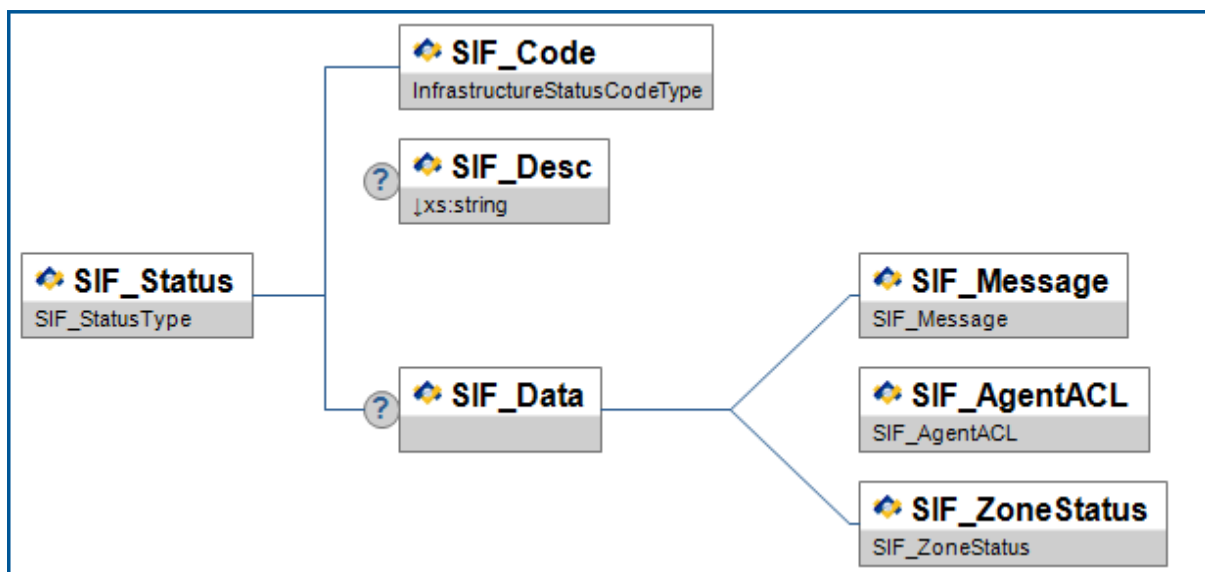


Figure 5.1.9-1: SIF_Status

Element/@Attribute	Char	Description	Type
SIF_Status		This element is used to signal a successful response.	
SIF_Code	M		InfrastructureStatusCodeType
SIF_Desc	O	An optional textual description/equivalent of SIF_Code.	xs:string xs:maxLength 1024

Element/@Attribute	Char	Description	Type
SIF_Data	O	Optional element to hold data related to a successful operation. This data is currently limited to a SIF_Message returned by the ZIS in response to a Pull-Mode Agent's SIF_GetMessage, SIF_AgentACL returned by the ZIS in response to SIF_Register and SIF_GetAgentACL, and SIF_ZoneStatus returned by the ZIS in response to SIF_GetZoneStatus.	choice of: SIF_Message SIF_AgentACL SIF_ZoneStatus

Table 5.1.9-1: SIF_Status

5.1.10 SIF_Error

This element is used to signal an unsuccessful response.

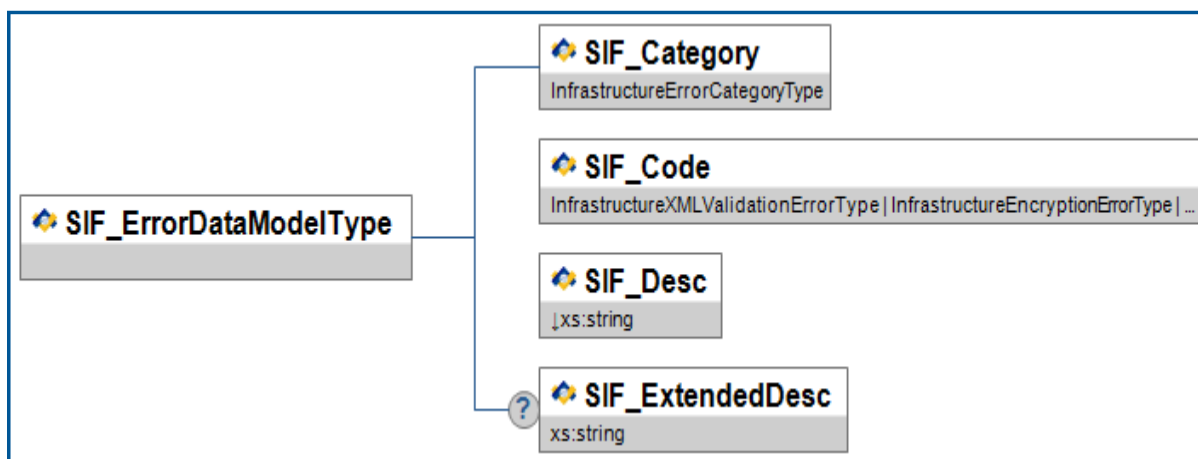


Figure 5.1.10-1: SIF_Error

Element/@Attribute	Char	Description	Type
SIF_Error		This element is used to signal an unsuccessful response.	
SIF_Category	M		InfrastructureErrorCategoryType
SIF_Code	M	See Error Codes .	union of: InfrastructureXMLValidationErrorType InfrastructureEncryptionErrorType InfrastructureAuthenticationErrorType InfrastructureAccessAndPermissionErrorType InfrastructureRegistrationErrorType InfrastructureProvisionErrorType InfrastructureSubscriptionErrorType InfrastructureRequestAndResponseErrorType InfrastructureEventReportingAndProcessingErrorType InfrastructureTransportErrorType InfrastructureSystemErrorType InfrastructureGenericMessageHandlingErrorType xs:token

Element/@Attribute	Char	Description	Type
SIF_Desc	M	A simple, easy to understand, description of the error. The primary consumer of this message is the application user. Example: "Unable to open database."	<code>xs:string</code> <div><code>xs:maxLength</code> 1024</div>
SIF_ExtendedDesc	O	An optional error description that is more complete and technical in nature. It is to be used as a diagnostic message in trouble-shooting procedures. Example: "The 'Students' table is opened in exclusive mode by user 'ADM1' (dbm.cpp, line 300)."	<code>xs:string</code>

Table 5.1.10-1: SIF_Error

5.1.11 SIF_Query

SIF's default query mechanism.

Note: With SIF Implementation Specification version 2.5 the choice element depicted in the diagram below has changed. In the XSD files, the choice element used to be rendered as a sequence of optional elements. Now, in the XSD files, the choice element is rendered as an `xs:choice`. This may cause problems with agents that treat the choice element as two optional elements rather than a choice of one and only one of the choices.

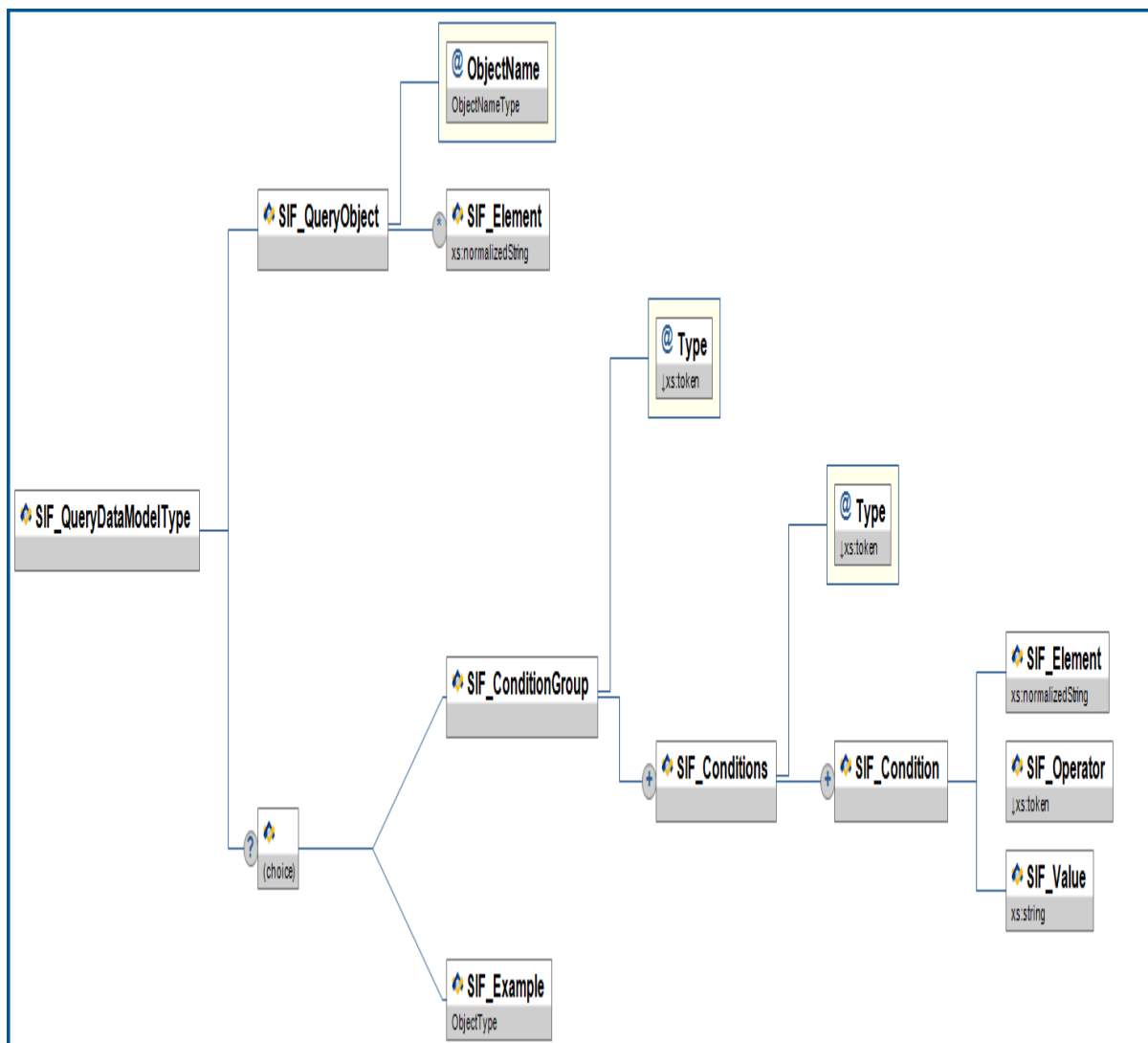


Figure 5.1.11-1: SIF_Query

Element/@ Attribute	Char	Description	Type
SIF_Query		<p>SIF's default query mechanism.</p> <p>Note: With SIF Implementation Specification version 2.5 the choice element depicted in the diagram below has changed. In the XSD files, the choice element used to be rendered as a sequence of optional elements. Now, in the XSD files, the choice element is rendered as an xs:choice. This may cause problems with agents that treat the choice element as two optional elements rather than a choice of one and only one of the choices.</p>	
SIF_QueryObject	M	This is the object that is being queried for.	
@ ObjectName	M	The name of the SIF object that is being queried for.	ObjectNameType

Element/@Attribute	Char	Description	Type
SIF_QueryObject/SIF_Element	OR	<p>Individual elements/attributes being requested of matching object. See SIF_Element Syntax below. If specified, only the elements/attributes requested are returned in the SIF_Response (with any parent elements/attributes); otherwise, all elements supported by the provider's object are returned.</p> <p>Note that this is a means to filter or select a subset of elements/attributes from a matching object; specifying elements/attributes here that do not occur in or are not supported in a matching object does not exclude that matching object from being returned. Include any existing parent elements/attributes of the elements/attributes that are requested but not present.</p>	xs:normalizedString
SIF_ConditionGroup	C	<p>Either SIF_ConditionGroup or SIF_Example may optionally be specified to present conditions matching objects should satisfy.</p> <p>SIF_ConditionGroup represents the conditions that the queried object(s) must meet. If conditions are specified, only those objects that meet the conditions are returned; otherwise, all objects of the specified name are returned.</p>	
@ Type	M	The Boolean operator for joining conditions (SIF_Conditions elements) within this element. Note that None should be used if there is only one SIF_Conditions element.	values: And Or None
SIF_ConditionGroup/SIF_Conditions	MR	This construct allows for nested conditions.	
@ Type	M	The boolean operator for joining conditions (SIF_Condition elements) within this element. Note that None should be used if there is only one SIF_Condition element.	values: And Or None
SIF_ConditionGroup/SIF_Conditions/SIF_Condition	MR	This element represents an individual condition.	
SIF_ConditionGroup/SIF_Conditions/SIF_Condition/SIF_Element	M	This is the element/attribute being queried. See below for syntax.	xs:normalizedString

Element/@Attribute	Char	Description	Type
SIF_ConditionGroup/SIF_Conditions/ SIF_Condition/SIF_Operator	M	The comparison operator for the condition.	values: EQ Equals LT Less Than GT Greater Than LE Less Than Or Equals GE Greater Than Or Equals NE Not Equals
SIF_ConditionGroup/SIF_Conditions/ SIF_Condition/SIF_Value	M	SIF_Value is the data that is used to compare with the value of the element or attribute.	xs:string
SIF_Example	C	An example SIF object that serves as a template for matching objects. There is an implied EQ operator for every element/attribute value present and an implied And group of all resulting conditions. Currently this is an experimental feature and limited to use with select objects; wider use may be considered in future versions of this specification.	ObjectType

Table 5.1.11-1: SIF_Query

5.1.11.1 SIF_ConditionGroup

The SIF_Query element may have a SIF_ConditionGroup element that may have one or more SIF_Conditions elements. A SIF_Conditions element may contain one or more SIF_Condition elements. Each SIF_Condition element defines a search criterion, which contains the following sub-elements. For example, if you wished to request the LibraryPatronStatus object for all teachers, the SIF_ConditionGroup would be: For example, if you wished to request the LearnerExclusion object for a student, the SIF_ConditionGroup would be:

```
<SIF_ConditionGroup Type="None">
  <SIF_Conditions Type="None">
    <SIF_Condition>
      <SIF_Element>@SIF_RefObject</SIF_Element>
      <SIF_Operator>EQ</SIF_Operator>
      <SIF_Value>StaffPersonal</SIF_Value>
    </SIF_Condition>
  </SIF_Conditions>
</SIF_ConditionGroup>
```

Example 5.1.11.1-1

If you wished to request the LibraryPatronStatus object for a specific teacher then the SIF_ConditionGroup would be:

```

<SIF_ConditionGroup Type="None">
  <SIF_Conditions Type="And">
    <SIF_Condition>
      <SIF_Element>@SIF_RefObject</SIF_Element>
      <SIF_Operator>EQ</SIF_Operator>
      <SIF_Value>StaffPersonal</SIF_Value>
    </SIF_Condition>
    <SIF_Condition>
      <SIF_Element>@SIF_RefId</SIF_Element>
      <SIF_Operator>EQ</SIF_Operator>
      <SIF_Value>D3E34B359D75101A8C3D00AA001A1652</SIF_Value>
    </SIF_Condition>
  </SIF_Conditions>
</SIF_ConditionGroup>

```

Example 5.1.11.1-2

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Request>
    <SIF_Header>
      <SIF_MsgId>A3E90785EFDA330DACB00785EFDA330D</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_Version>2.*</SIF_Version>
    <SIF_MaxBufferSize>1048576</SIF_MaxBufferSize>
    <SIF_Query>
      <SIF_QueryObject ObjectName="StudentPersonal" />
      <SIF_ConditionGroup Type="None">
        <SIF_Conditions Type="None">
          <SIF_Condition>
            <SIF_Element>Name/LastName</SIF_Element>
            <SIF_Operator>EQ</SIF_Operator>
            <SIF_Value>Smith</SIF_Value>
          </SIF_Condition>
        </SIF_Conditions>
      </SIF_ConditionGroup>
    </SIF_Query>
  </SIF_Request>
</SIF_Message>

```

Example 5.1.11.1-3: SIF_ConditionGroup querying into an object

5.1.11.2 SIF_Element Syntax

To reference individual elements/attributes in query criteria for objects, and in lists of individual elements/attributes to be returned from matching objects, SIF defines a path syntax which is based on a small subset of

[XPath], for use in SIF_Element. Elements are specified by name (e.g. Name) and attributes are specified by name, prefixed with @ (e.g. @Type). Namespace prefixes may precede element/attribute names as necessary (e.g. @xml:lang) and reference the current prefix-to-namespace mappings within the XML of the request. To reference child elements or attributes of child elements, a path notation is used where each element/attribute in the path is separated by / (e.g. Name/FirstName, Name/@Type). The object's element is the root element and is not included when referencing child elements (e.g. Name/FirstName, not StudentPersonal/Name/FirstName); no / is required when referencing attributes of the object itself (e.g. @RefId, not StudentPersonal/@RefId).

SIF_Condition/SIF_Element may also contain XPath predicates (e.g. [@Type='04']) to allow for more precise matching, especially with regard to repeatable elements with "key" attributes. The following SIF_Condition would match object with any FirstName of Cameron:

```

<SIF_Condition>
  <SIF_Element>Name/FirstName</SIF_Element>
  <SIF_Operator>EQ</SIF_Operator>
  <SIF_Value>Cameron</SIF_Value>
</SIF_Condition>

```

Example 5.1.11.2-1

Using a predicate allows the requester to specifically query the person's name of record (04) vs. his/her previous, professional, current legal name, etc.

```
<SIF_Condition>
  <SIF_Element>Name[@Type='04']/FirstName</SIF_Element>
  <SIF_Operator>EQ</SIF_Operator>
  <SIF_Value>Cameron</SIF_Value>
</SIF_Condition>
```

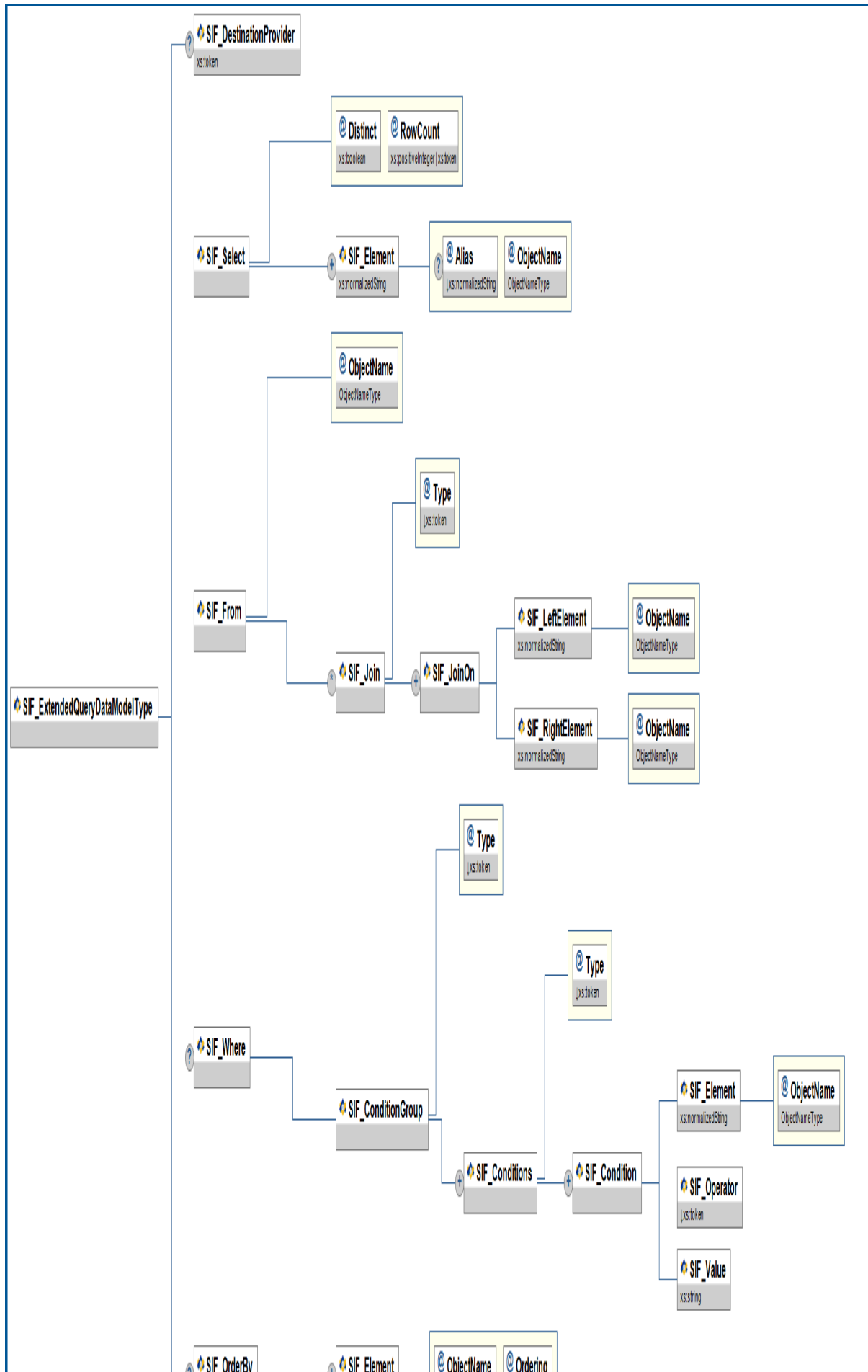
Example 5.1.11.2-2

Predicate expressions supported in SIF are limited to `or`, `and`, `=`, element/attribute names with optional prefixes and accessing nested elements/attributes using `/`.

5.1.12 SIF_ExtendedQuery

SIF's default query mechanism for `SIF_Request`, `SIF_Query`, has several limitations that limit its usefulness when creating reporting applications that process data from a SIF zone. `SIF_Query` is limited to matching only one object type per query, requiring applications to manually join together results as needed for reporting and general data processing. `SIF_ExtendedQuery` is designed to allow for joins on object identifiers/RefIds and to allow retrieval of data in a row/column fashion similar to SQL. Each returned column may contain hierarchical XML elements/objects. While envisioned as the primary mechanism for SIF-based `ReportManifests`, Providers and Responders in a Zone may support `SIF_ExtendedQuery` in addition to `SIF_Query`. Support for `SIF_ExtendedQuery` can be declared in and retrieved from the Zone in various Infrastructure messages and objects.

Note that matching rows are generated solely based on the `SIF_From` clause, with optional join criteria, optionally limited/filtered by the `SIF_Where` clause. If a repeatable element is requested as a column in `SIF_Select`, this does not generate multiple rows for each occurrence of matching elements; all elements are returned in the corresponding column within a single row.



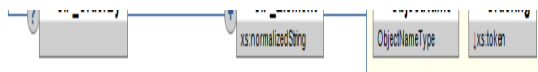


Figure 5.1.12-1: SIF_ExtendedQuery

Element/@Attribute	Char	Description	Type
SIF_ExtendedQuery		<p>SIF's default query mechanism for <code>SIF_Request</code>, <code>SIF_Query</code>, has several limitations that limit its usefulness when creating reporting applications that process data from a SIF zone. <code>SIF_Query</code> is limited to matching only one object type per query, requiring applications to manually join together results as needed for reporting and general data processing. <code>SIF_ExtendedQuery</code> is designed to allow for joins on object identifiers/RefsIds and to allow retrieval of data in a row/column fashion similar to SQL. Each returned column may contain hierarchical XML elements/objects. While envisioned as the primary mechanism for SIF-based <code>ReportManifests</code>, <code>Providers</code> and <code>Responders</code> in a Zone may support <code>SIF_ExtendedQuery</code> in addition to <code>SIF_Query</code>. Support for <code>SIF_ExtendedQuery</code> can be declared in and retrieved from the Zone in various Infrastructure messages and objects.</p> <p>Note that matching rows are generated solely based on the <code>SIF_From</code> clause, with optional join criteria, optionally limited/filtered by the <code>SIF_Where</code> clause. If a repeatable element is requested as a column in <code>SIF_Select</code>, this does not generate multiple rows for each occurrence of matching elements; all elements are returned in the corresponding column within a single row.</p>	
<code>SIF_DestinationProvider</code>	O	If no <code>SIF_DestinationId</code> applies to the request and this element is supplied, the Requester specifies that the extended query be routed to the Provider on record for the given object name. If this element is omitted and no <code>SIF_DestinationId</code> applies to the request, the ZIS routes the request to the Provider on record for the object name in <code>SIF_From</code> .	<code>xs:token</code>
<code>SIF_Select</code>	M	Identifies which data elements/attribute are to be returned as columns in each matching row.	
@ <code>Distinct</code>	M	Specifies whether query results should return all rows (<code>false</code>) or just distinct ones (<code>true</code>). Rows are distinct if at least one column differs between them.	<code>xs:boolean</code>
@ <code>RowCount</code>	M	The maximum number of rows to return. If <code>All</code> , return all rows, otherwise return the top rows up to the maximum row count indicated.	union of: <code>xs:positiveInteger</code> additional values: <code>All</code>

	Element/@Attribute	Char	Description	Type
	SIF_Select/SIF_Element	MR	Indicates the element/attribute to return as a column. Contents can be left empty to return the whole object specified in <code>ObjectName</code> , or * can be designated to return all attributes and immediate child elements of the object specified in <code>ObjectName</code> , or <code>SIF_Element Syntax</code> can be specified, relative to the object specified in <code>ObjectName</code> . Requested attributes are to be returned as the text value of the corresponding attribute, elements as a copy of the XML element itself including attributes if they exist.	<code>xs:normalizedString</code>
@	Alias	O	Optional caption for the column.	<code>xs:normalizedString</code> <div><code>xs:maxLength</code> 64</div>
@	ObjectName	M	The name of the object from which to retrieve element/attributes.	<code>ObjectNameType</code>
	SIF_From	M	Join specification for the query if more than one object is being queried. If only one object is being queried, specify it without the <code>SIF_Join</code> clause. This clause generates the matching rows returned, optionally limited/filtered by the <code>SIF_Where</code> clause. Each object referenced in the <code>SIF_Select</code> , <code>SIF_Where</code> and <code>SIF_OrderBy</code> clauses must be included here.	
@	ObjectName	M	The name of the object to query.	<code>ObjectNameType</code>
	SIF_From/SIF_Join	OR	Additional objects to query, with join conditions specifying the relationships between objects.	
@	Type	M	Type of relational join.	values: <code>Inner</code> <code>LeftOuter</code> <code>RightOuter</code> <code>FullOuter</code>
	SIF_From/SIF_Join/SIF_JoinOn	MR	Specifies the conditions for the join.	
	SIF_From/SIF_Join/SIF_JoinOn/ SIF_LeftElement	M	Specifies the left-side element/attribute on which to constrain the join. Currently only support for keys/RefIds/RefId references is required.	<code>xs:normalizedString</code>
@	ObjectName	M	Name of the object that contains the element/attribute.	<code>ObjectNameType</code>

	Element/@Attribute	Char	Description	Type
	SIF_From/SIF_Join/SIF_JoinOn/ SIF_RightElement	M	Specifies right left-side element/attribute on which to constrain the join. Currently only support for keys/RefIds/RefId references is required.	<code>xs:normalizedString</code>
@	ObjectName	M	Name of the object that contains the element/attribute.	<code>ObjectNameType</code>
	SIF_Where	O	Optionally specifies conditions to limit/filter rows resulting from the <code>SIF_From</code> clause.	
	SIF_Where/SIF_ConditionGroup	M	Conditions that matching rows must meet.	
@	Type	M	The Boolean operator for joining conditions (<code>SIF_Conditions</code> elements) within this element. Note that <code>None</code> should be used if there is only one <code>SIF_Conditions</code> element.	values: <code>And</code> <code>Or</code> <code>None</code>
	SIF_Where/SIF_ConditionGroup/ SIF_Conditions	MR	This construct allows for nested conditions.	
@	Type	M	The boolean operator for joining conditions (<code>SIF_Condition</code> elements) within this element. Note that <code>None</code> should be used if there is only one <code>SIF_Condition</code> element.	values: <code>And</code> <code>Or</code> <code>None</code>
	SIF_Where/SIF_ConditionGroup/ SIF_Conditions/SIF_Condition	MR	This element represents an individual condition.	
	SIF_Where/SIF_ConditionGroup/ SIF_Conditions/SIF_Condition/ SIF_Element	M	This is the element/attribute being queried.	<code>xs:normalizedString</code>
@	ObjectName	M	The name of the object containing the element/attribute.	<code>ObjectNameType</code>
	SIF_Where/SIF_ConditionGroup/ SIF_Conditions/SIF_Condition/ SIF_Operator	M	The comparison operator for the condition.	values: <code>EQ</code> Equals <code>LT</code> Less Than <code>GT</code> Greater Than <code>LE</code> Less Than Or Equals <code>GE</code> Greater Than Or Equals <code>NE</code> Not Equals

	Element/@Attribute	Char	Description	Type
	SIF_Where/SIF_ConditionGroup/ SIF_Conditions/SIF_Condition/ SIF_Value	M	SIF_Value is the data that is used to compare with the value of the element or attribute.	xs:string
	SIF_OrderBy	O	An optional list of elements/attributes by which to sort the resulting rows.	
	SIF_OrderBy/SIF_Element	MR	Indicates the element/attribute by which to sort.	xs:normalizedString
@	ObjectName	M	The name of the object containing the element/attribute.	ObjectNameType
@	Ordering	M	Whether to order the element/attribute in ascending or descending order.	values: Ascending Descending

Table 5.1.12-1: SIF_ExtendedQuery

```

<SIF_ExtendedQuery>
  <SIF_Select Distinct="false" RowCount="All">
    <SIF_Element ObjectName="StudentPersonal" />
  </SIF_Select>
  <SIF_From ObjectName="StudentPersonal" />
</SIF_ExtendedQuery>

```

Example 5.1.12-1: Selecting all StudentPersonal objects

```

<SIF_ExtendedQuery>
  <SIF_Select Distinct="false" RowCount="All">
    <SIF_Element ObjectName="StudentPersonal">*</SIF_Element>
  </SIF_Select>
  <SIF_From ObjectName="StudentPersonal" />
</SIF_ExtendedQuery>

```

Example 5.1.12-3: Selecting all attributes and immediate child elements of StudentPersonal as columns from all StudentPersonal objects

```

<SIF_ExtendedQuery>
  <SIF_Select Distinct="false" RowCount="All">
    <SIF_Element ObjectName="StudentPersonal">@RefId</SIF_Element>
    <SIF_Element ObjectName="StudentPersonal">Name/FirstName</SIF_Element>
    <SIF_Element ObjectName="StudentPersonal">Name/LastName</SIF_Element>
    <SIF_Element ObjectName="StudentPersonal">EmailList</SIF_Element>
  </SIF_Select>
  <SIF_From ObjectName="StudentPersonal" />
</SIF_ExtendedQuery>

```

Example 5.1.12-5: Selecting specific attributes and elements from all StudentPersonal objects

```

<SIF_ExtendedQuery>
  <SIF_Select Distinct="true" RowCount="All">
    <SIF_Element ObjectName="StudentPersonal" />
    <SIF_Element ObjectName="StudentSchoolEnrollment" Alias="Student Entry Date">EntryDate</SIF_Element>
  </SIF_Select>
  <SIF_From ObjectName="StudentPersonal">
    <SIF_Join Type="Inner">
      <SIF_JoinOn>
        <SIF_LeftElement ObjectName="StudentPersonal">@RefId</SIF_LeftElement>
        <SIF_RightElement ObjectName="StudentSchoolEnrollment">@StudentPersonalRefId</SIF_RightElement>
      </SIF_JoinOn>
    </SIF_Join>
  </SIF_From>
  <SIF_Where>
    <SIF_ConditionGroup Type="And">
      <SIF_Conditions Type="And">
        <SIF_Condition>
          <SIF_Element ObjectName="StudentSchoolEnrollment">@SchoolInfoRefId</SIF_Element>
          <SIF_Operator>EQ</SIF_Operator>
          <SIF_Value>A3E90785EFDA330DACB00785EFDA330D</SIF_Value>
        </SIF_Condition>
        <SIF_Condition>
          <SIF_Element ObjectName="StudentSchoolEnrollment">@SchoolYear</SIF_Element>
          <SIF_Operator>EQ</SIF_Operator>
          <SIF_Value>2007</SIF_Value>
        </SIF_Condition>
        <SIF_Condition>
          <SIF_Element ObjectName="StudentSchoolEnrollment">@MembershipType</SIF_Element>
          <SIF_Operator>EQ</SIF_Operator>
          <SIF_Value>Home</SIF_Value>
        </SIF_Condition>
      </SIF_Conditions>
      <SIF_Conditions Type="Or">
        <SIF_Condition>
          <SIF_Element ObjectName="StudentSchoolEnrollment">@TimeFrame</SIF_Element>
          <SIF_Operator>EQ</SIF_Operator>
          <SIF_Value>Current</SIF_Value>
        </SIF_Condition>
        <SIF_Condition>
          <SIF_Element ObjectName="StudentSchoolEnrollment">@TimeFrame</SIF_Element>
          <SIF_Operator>EQ</SIF_Operator>
          <SIF_Value>Future</SIF_Value>
        </SIF_Condition>
      </SIF_Conditions>
    </SIF_ConditionGroup>
  </SIF_Where>
  <SIF_OrderBy>
    <SIF_Element ObjectName="StudentPersonal" Ordering="Ascending">Name/LastName</SIF_Element>
  </SIF_OrderBy>
</SIF_ExtendedQuery>

```

Example 5.1.12-7: Selecting StudentPersonal objects along with each student's EntryDate from StudentSchoolEnrollment for a specific school, school year and other StudentSchoolEnrollment values, sorted by student's last name

```

<SIF_ExtendedQuery>
  <SIF_Select Distinct="false" RowCount="All">
    <SIF_Element ObjectName="StudentPersonal" />
    <SIF_Element ObjectName="StudentSchoolEnrollment" />
    <SIF_Element ObjectName="SchoolInfo">SchoolName</SIF_Element>
  </SIF_Select>
  <SIF_From ObjectName="StudentPersonal">
    <SIF_Join Type="Inner">
      <SIF_JoinOn>
        <SIF_LeftElement ObjectName="StudentPersonal">@RefId</SIF_LeftElement>
        <SIF_RightElement ObjectName="StudentSchoolEnrollment">@StudentPersonalRefId</SIF_RightElement>
      </SIF_JoinOn>
    </SIF_Join>
    <SIF_Join Type="Inner">
      <SIF_JoinOn>
        <SIF_LeftElement ObjectName="StudentSchoolEnrollment">@SchoolInfoRefId</SIF_LeftElement>
        <SIF_RightElement ObjectName="SchoolInfo">@RefId</SIF_RightElement>
      </SIF_JoinOn>
    </SIF_Join>
  </SIF_From>
  <SIF_Where>
    <SIF_ConditionGroup Type="None">
      <SIF_Conditions Type="None">
        <SIF_Condition>
          <SIF_Element ObjectName="StudentPersonal">@RefId</SIF_Element>
          <SIF_Operator>EQ</SIF_Operator>
          <SIF_Value>12345678901234567890123456789012</SIF_Value>
        </SIF_Condition>
      </SIF_Conditions>
    </SIF_ConditionGroup>
  </SIF_Where>
</SIF_ExtendedQuery>

```

Example 5.1.12-9: Selecting a specific StudentPersonal's StudentSchoolEnrollment objects, along with the corresponding school name for each enrollment

5.1.12.1 Mapping SIF_Query to SIF_ExtendedQuery

While there are differences in how matching objects are returned, note that all non-SIF_Example SIF_Query-based requests can be mapped to a corresponding SIF_ExtendedQuery-based request:

1	Place SIF_Query/SIF_QueryObject/@ObjectName in SIF_ExtendedQuery/SIF_From/@ObjectName.
2	If elements/attributes are specified in SIF_Query/SIF_QueryObject/SIF_Element, place them in SIF_ExtendedQuery/SIF_Select/SIF_Element with @ObjectName set to SIF_Query/SIF_QueryObject/@ObjectName. Otherwise in SIF_Select, specify an empty SIF_Element element with @ObjectName set to SIF_Query/SIF_QueryObject/@ObjectName.
3	If SIF_Query/SIF_ConditionGroup exists, place it in SIF_ExtendedQuery/SIF_Where setting @ObjectName to SIF_Query/SIF_QueryObject/@ObjectName in every occurrence of SIF_Element.
4	Set SIF_Select/@Distinct to false and SIF_Select/@RowCount to All.

Table 5.1.12.1-1: Mapping SIF_Query to SIF_ExtendedQuery

```

<SIF_Query>
  <SIF_QueryObject ObjectName="StudentPersonal">
    <SIF_Element>Name/FirstName</SIF_Element>
    <SIF_Element>Name/LastName</SIF_Element>
  </SIF_QueryObject>
  <SIF_ConditionGroup Type="None">
    <SIF_Conditions Type="None">
      <SIF_Condition>
        <SIF_Element>@RefId</SIF_Element>
        <SIF_Operator>EQ</SIF_Operator>
        <SIF_Value>F0F29E6AE742498D9CB91CBB3BE6890E</SIF_Value>
      </SIF_Condition>
    </SIF_Conditions>
  </SIF_ConditionGroup>
</SIF_Query>

```

Example 5.1.12.1-1: Input SIF_Query

```

<SIF_ExtendedQuery>
  <SIF_Select Distinct="false" RowCount="All">
    <SIF_Element ObjectName="StudentPersonal">Name/FirstName</SIF_Element>
    <SIF_Element ObjectName="StudentPersonal">Name/LastName</SIF_Element>
  </SIF_Select>
  <SIF_From ObjectName="StudentPersonal" />
  <SIF_Where>
    <SIF_ConditionGroup Type="None">
      <SIF_Conditions Type="None">
        <SIF_Condition>
          <SIF_Element ObjectName="StudentPersonal">@RefId</SIF_Element>
          <SIF_Operator>EQ</SIF_Operator>
          <SIF_Value>F0F29E6AE742498D9CB91CBB3BE6890E</SIF_Value>
        </SIF_Condition>
      </SIF_Conditions>
    </SIF_ConditionGroup>
  </SIF_Where>
</SIF_ExtendedQuery>

```

Example 5.1.12.1-2: Corresponding SIF_ExtendedQuery

5.1.13 SIF_ExtendedQueryResults

This element provides a wrapper for data returned in response to a SIF_ExtendedQuery. Used in SIF_Response and SIF_ReportObject.

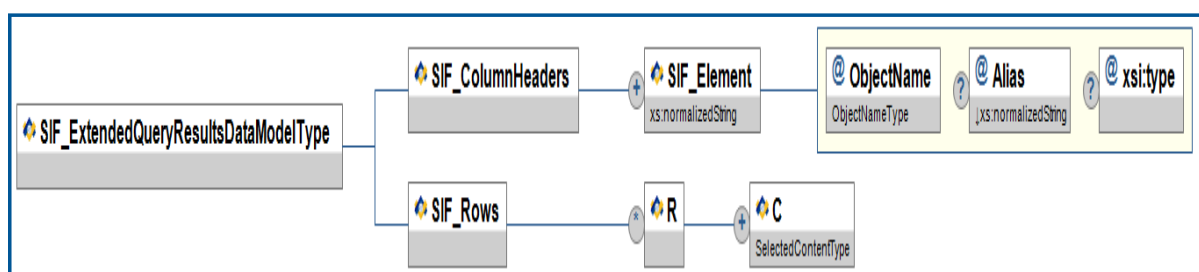


Figure 5.1.13-1: SIF_ExtendedQueryResults

Element/@ Attribute	Char	Description	Type
SIF_ExtendedQueryResults		This element provides a wrapper for data returned in response to a SIF_ExtendedQuery. Used in SIF_Response and SIF_ReportObject.	
SIF_ColumnHeaders	M	Provides the element/attribute and caption information for each column supplied in SIF_ExtendedQuery. The order must correspond to the order of the elements as requested in SIF_ExtendedQuery.	
SIF_ColumnHeaders/SIF_Element	MR	The element/attribute specified for the column in SIF_ExtendedQuery.	xs:normalizedString
@ ObjectName	M	The object in which the element/attribute occurs.	ObjectNameType
@ Alias	O	The caption for the column, if specified in SIF_ExtendedQuery.	xs:normalizedString <div>xs:maxLength 64</div>

	Element/@ Attribute	Char	Description	Type
@	xsi:type	O	Optionally allows type of column value to be explicitly communicated.	
	SIF_Rows	M	A list of matching rows resulting from the supplied SIF_ExtendedQuery. Note that the complete list of rows may span multiple SIF_Response messages, per the SIF_MaxBufferSize supplied in SIF_Request. If there are no matching rows, this is an empty list.	
	SIF_Rows/R	OR	An individual matching row resulting from the supplied SIF_ExtendedQuery.	
	SIF_Rows/R/C	MR	Contains the value of each column specified in SIF_ExtendedQuery/SIF_Select. The order of the columns must correspond to the order of the elements as requested in SIF_ExtendedQuery. Note the number of columns may be expanded from the requested columns if * is indicated one or more times in the SIF_Select clause.	SelectedContentType

Table 5.1.13-1: SIF_ExtendedQueryResults

```

<SIF_ExtendedQueryResults>
  <SIF_ColumnHeaders>
    <SIF_Element ObjectName="StudentPersonal" />
    <SIF_Element ObjectName="StudentSchoolEnrollment" />
    <SIF_Element ObjectName="SchoolInfo">SchoolName</SIF_Element>
  </SIF_ColumnHeaders>
  <SIF_Rows>
    <R>
      <C>
        <StudentPersonal RefId="12345678901234567890123456789012">
          <!--...-->
        </StudentPersonal>
      </C>
      <C>
        <StudentSchoolEnrollment RefId="AED4AEF825284D7E9F082EBBEB1999FA" StudentPersonalRefId="12345678901234567890123456789012">
          <!--...-->
        </StudentSchoolEnrollment>
      </C>
      <C>
        <SchoolName>SIFA High</SchoolName>
      </C>
    </R>
    <R>
      <C>
        <StudentPersonal RefId="12345678901234567890123456789012">
          <!--...-->
        </StudentPersonal>
      </C>
      <C>
        <StudentSchoolEnrollment RefId="AED4AEF825284D7E9F082EBBEB1999FA" StudentPersonalRefId="12345678901234567890123456789012">
          <!--...-->
        </StudentSchoolEnrollment>
      </C>
      <C>
        <SchoolName>SIFA University</SchoolName>
      </C>
    </R>
  </SIF_Rows>
</SIF_ExtendedQueryResults>

```

Example 5.1.13-1: SIF_ExtendedQueryResults

5.2 Messages

5.2.1 SIF_Ack

This message is used as an acknowledgement for infrastructure messages. All infrastructure messages will return a `SIF_Ack` as a result to indicate if the request was successful or not. A `SIF_Ack` must contain either a `SIF_Status` element acknowledging a successful result or a `SIF_Error` element describing the failure. The `SIF_Error` element contains a standardized error number as well as a description of the error.

A successful `SIF_Ack` is typically returned to the caller containing a `SIF_Header`, `SIF_OriginalSourceId`, `SIF_OriginalMsgId` and the `SIF_Status` element. In situations where additional information needs to be returned to the caller, a `SIF_Data` element can be added to the `SIF_Status` element.

In addition, successful `SIF_Ack` messages may also be sent to the ZIS under two conditions. The first is when a pull-mode agent requests that a message is to be removed from its queue. The second is when an agent which has invoked SMB wishes to end SMB handling. In that case, the agent sends a "Final" `SIF_Ack` to the ZIS. In each instance the ZIS returns a `SIF_Ack` in response to the agent's `SIF_Ack` message.

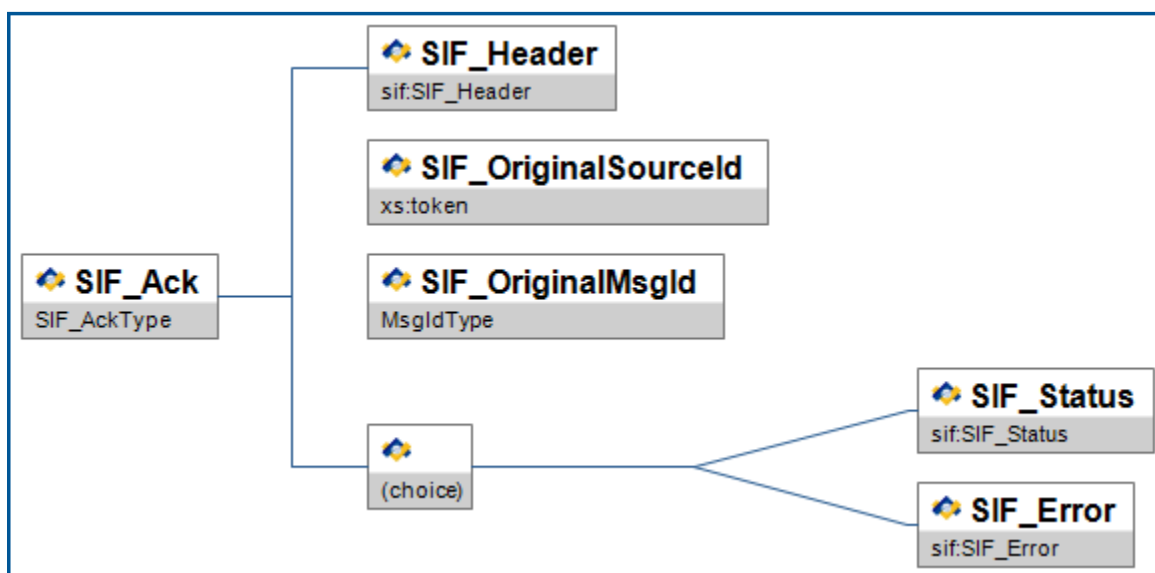


Figure 5.2.1-1: `SIF_Ack`

Element/@Attribute	Char	Description	Type
<code>SIF_Ack</code>	M	This message is used as an acknowledgement to an infrastructure message.	
<code>SIF_Header</code>	M	Header information associated with this message.	<code>SIF_Header</code>
<code>SIF_OriginalSourceId</code>	M	The <code>SIF_SourceId</code> of the infrastructure message for which the <code>SIF_Ack</code> serves as a response.	<code>xs:token</code>
<code>SIF_OriginalMsgId</code>	M	The <code>SIF_MsgId</code> of the infrastructure message for which the <code>SIF_Ack</code> message serves as a response.	<code>MsgIdType</code>

Element/@Attribute	Char	Description	Type
SIF_Status	C	This element is used to signal a successful response.	SIF_Status
SIF_Error	C	This element is used to signal an unsuccessful response.	SIF_Error

Table 5.2.1-1: SIF_Ack

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>AB1058CD3261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>SifInfo_TestZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseyLib</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>1298ACEF3261545A31905937B265CE01</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>0</SIF_Code>
    <SIF_Data>
      <SIF_Message Version="2.5">
        <SIF_Request>
          <SIF_Header>
            <SIF_MsgId>A3E90785EFDA330DACB00785EFDA330D</SIF_MsgId>
            <SIF_Timestamp>2006-02-18T08:39:02-08:00</SIF_Timestamp>
            <SIF_SourceId>RamseySIS</SIF_SourceId>
          </SIF_Header>
          <SIF_Version>2.*</SIF_Version>
          <SIF_MaxBufferSize>1048576</SIF_MaxBufferSize>
          <SIF_Query>
            <SIF_QueryObject ObjectName="LibraryPatronStatus" />
            <SIF_ConditionGroup Type="None">
              <SIF_Conditions Type="None">
                <SIF_Condition>
                  <SIF_Element>@SIF_RefObject</SIF_Element>
                  <SIF_Operator>EQ</SIF_Operator>
                  <SIF_Value>StaffPersonal</SIF_Value>
                </SIF_Condition>
              </SIF_Conditions>
            </SIF_ConditionGroup>
          </SIF_Query>
        </SIF_Request>
      </SIF_Message>
    </SIF_Data>
  </SIF_Status>
</SIF_Ack>
</SIF_Message>

```

Example 5.2.1-1: SIF_Ack Status Message

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>CD5087FE3261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>1945CD783261545A31905937B265CE01</SIF_OriginalMsgId>
    <SIF_Error>
      <SIF_Category>3</SIF_Category>
      <SIF_Code>5</SIF_Code>
      <SIF_Desc>Sender's certificate is not trusted</SIF_Desc>
      <SIF_ExtendedDesc>Agent requires certificate issued by ISD11 CA</SIF_ExtendedDesc>
    </SIF_Error>
  </SIF_Ack>
</SIF_Message>

```

Example 5.2.1-4: SIF_Ack Error Message

5.2.2 SIF_Event

`SIF_Event` is used to deliver event objects as defined in SIF. Events represent the availability of a new data object, changes to, or deletions of data object.

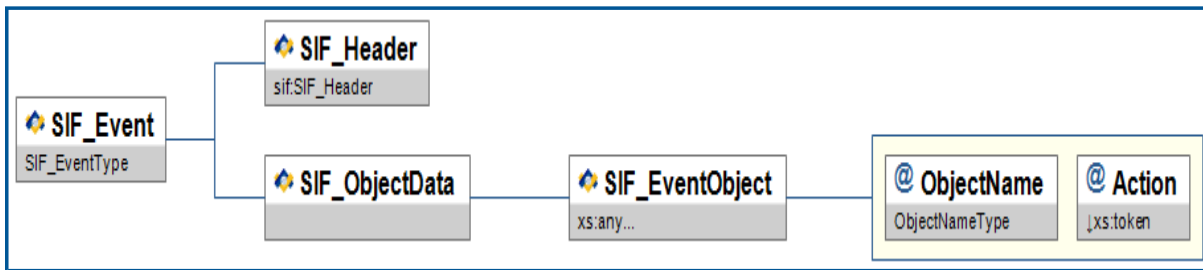


Figure 5.2.2-1: `SIF_Event`

Element/@Attribute	Char	Description	Type
SIF_Event	M	<code>SIF_Event</code> is used to deliver event objects as defined in SIF.	
SIF_Header	M	Header information associated with this message.	<code>SIF_Header</code>
SIF_ObjectData	M		
SIF_ObjectData/SIF_EventObject	M	Contains the object (partial or whole) that is being added, changed or deleted.	<code><xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" namespace="http://www.sif-implementation.org/1.5/data"></code>
@ObjectName	M	This is the name of the object being added, changed or deleted.	<code>ObjectNameType</code>

	Element/@Attribute	Char	Description	Type
@	Action	M	<p>This is the action associated with the object that is being conveyed by this SIF_Event.</p>	<p>values:</p> <p>Add Delete Change</p>

Table 5.2.2-1: SIF_Event

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Event>
    <SIF_Header>
      <SIF_MsgId>AB34DC093261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_ObjectData>
      <SIF_EventObject ObjectName="StudentPersonal" Action="Change">
        <StudentPersonal RefId="D3E34B359D75101A8C3D00AA001A1652">
          <PhoneNumberList>
            <PhoneNumber Type="0096">
              <Number>(312) 555-1234</Number>
            </PhoneNumber>
          </PhoneNumberList>
        </StudentPersonal>
      </SIF_EventObject>
    </SIF_ObjectData>
  </SIF_Event>
</SIF_Message>
```

Example 5.2.2-1: SIF_Event Message with StudentPersonal changes

5.2.3 SIF_Provide

The SIF_Provide message is used to attempt registering as the provider of one or more data objects.

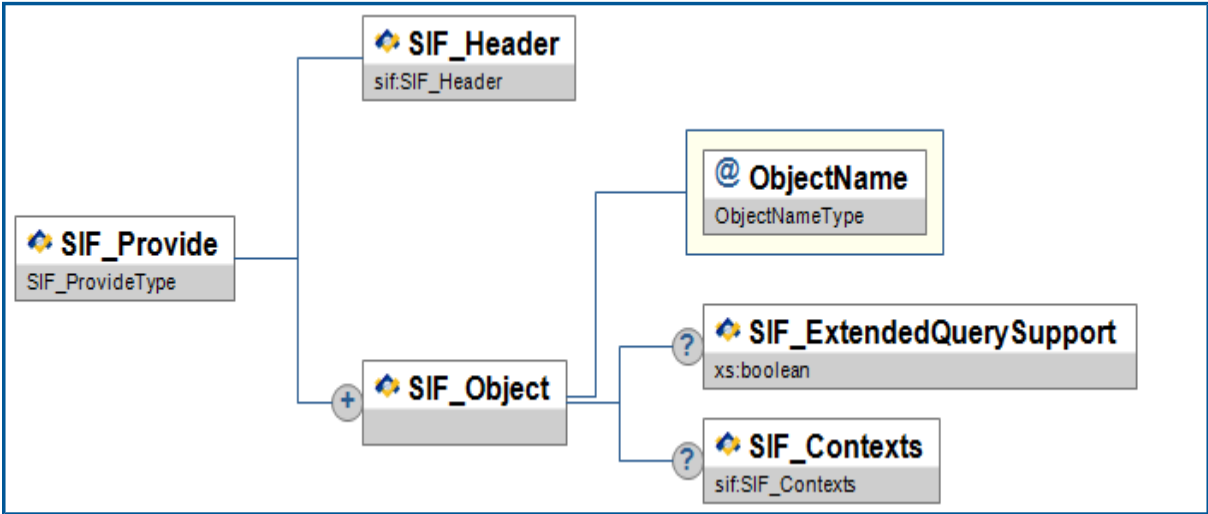


Figure 5.2.3-1: SIF_Provide

Element/@Attribute	Char	Description	Type
SIF_Provide	M	The <code>SIF_Provide</code> message is used for advertising the provision of data objects.	
SIF_Header	M	Header information associated with this message.	<code>SIF_Header</code>
SIF_Object	MR	This is the object that is being provided.	
@ ObjectName	M	The name of the SIF object that is being provided.	<code>ObjectNameType</code>
SIF_Object/SIF_ExtendedQuerySupport	O	Whether or not the Agent supports <code>SIF_ExtendedQuery</code> for this object.	<code>xs:boolean</code>
SIF_Object/SIF_Contexts	O	The contexts in which the object is being provided; if omitted, the context is <code>SIF_Default</code> .	<code>SIF_Contexts</code>

Table 5.2.3-1: *SIF_Provide*

```

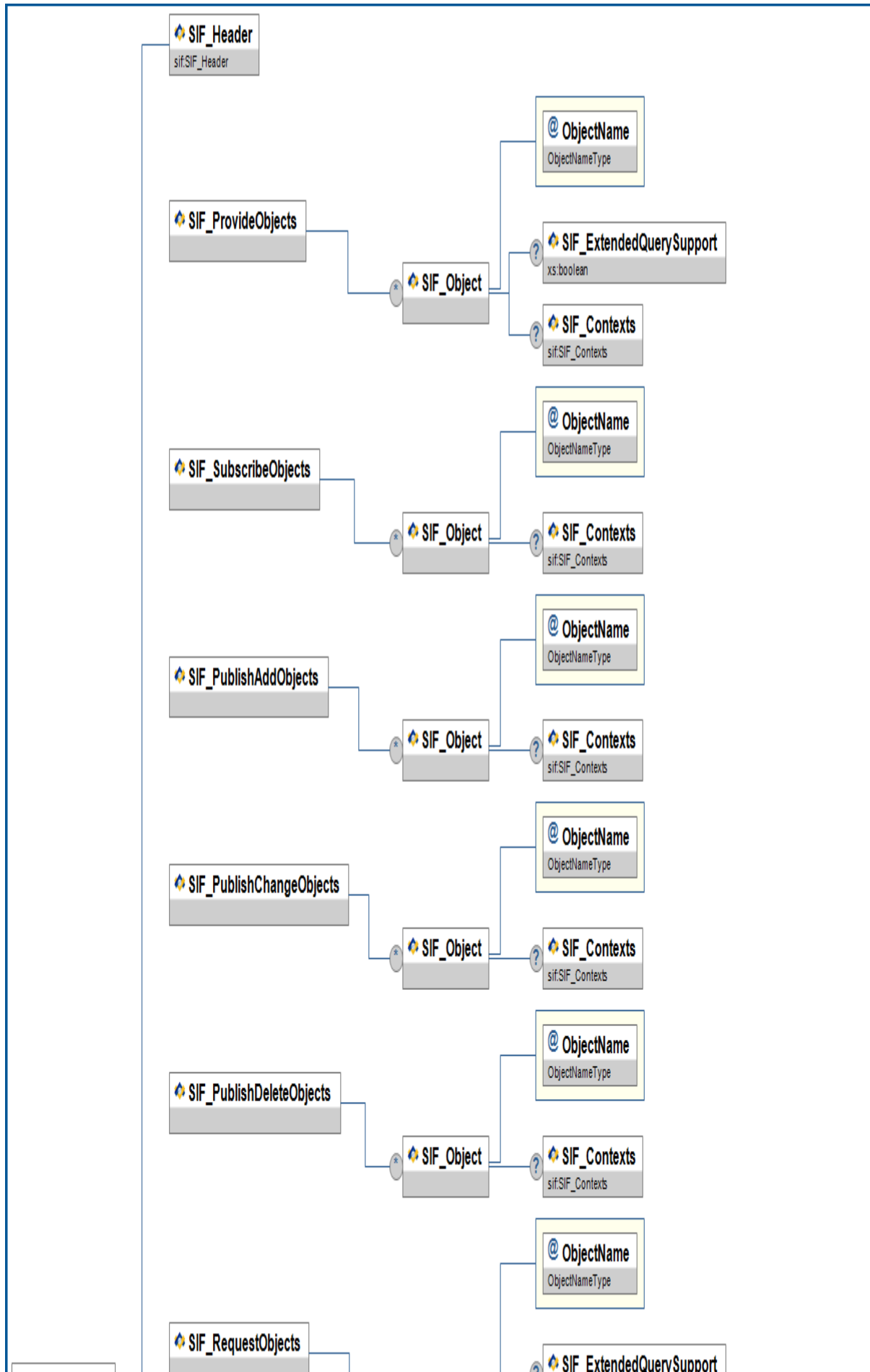
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Provide>
    <SIF_Header>
      <SIF_MsgId>34DC87FE3261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_Object ObjectName="StudentPersonal" />
    <SIF_Object ObjectName="StudentSchoolEnrollment" />
  </SIF_Provide>
</SIF_Message>

```

Example 5.2.3-1: *SIF_Provide*

5.2.4 SIF_Provision

Once registered, this message allows an agent to announce to the ZIS the functionality the agent will provide. The ZIS compares the functionality to its access control list and either returns a failure or a success. Upon success, the ZIS performs an atomic update of its provide/subscribe database entries for the agent to match the objects listed in this message and atomically updates other stored settings for the agent. A ZIS must not allow an agent to perform operations that it did not successfully announce. Agents should be aware that if the access control list changes after a successful `SIF_Provision`, some operations may still be rejected with access control errors.



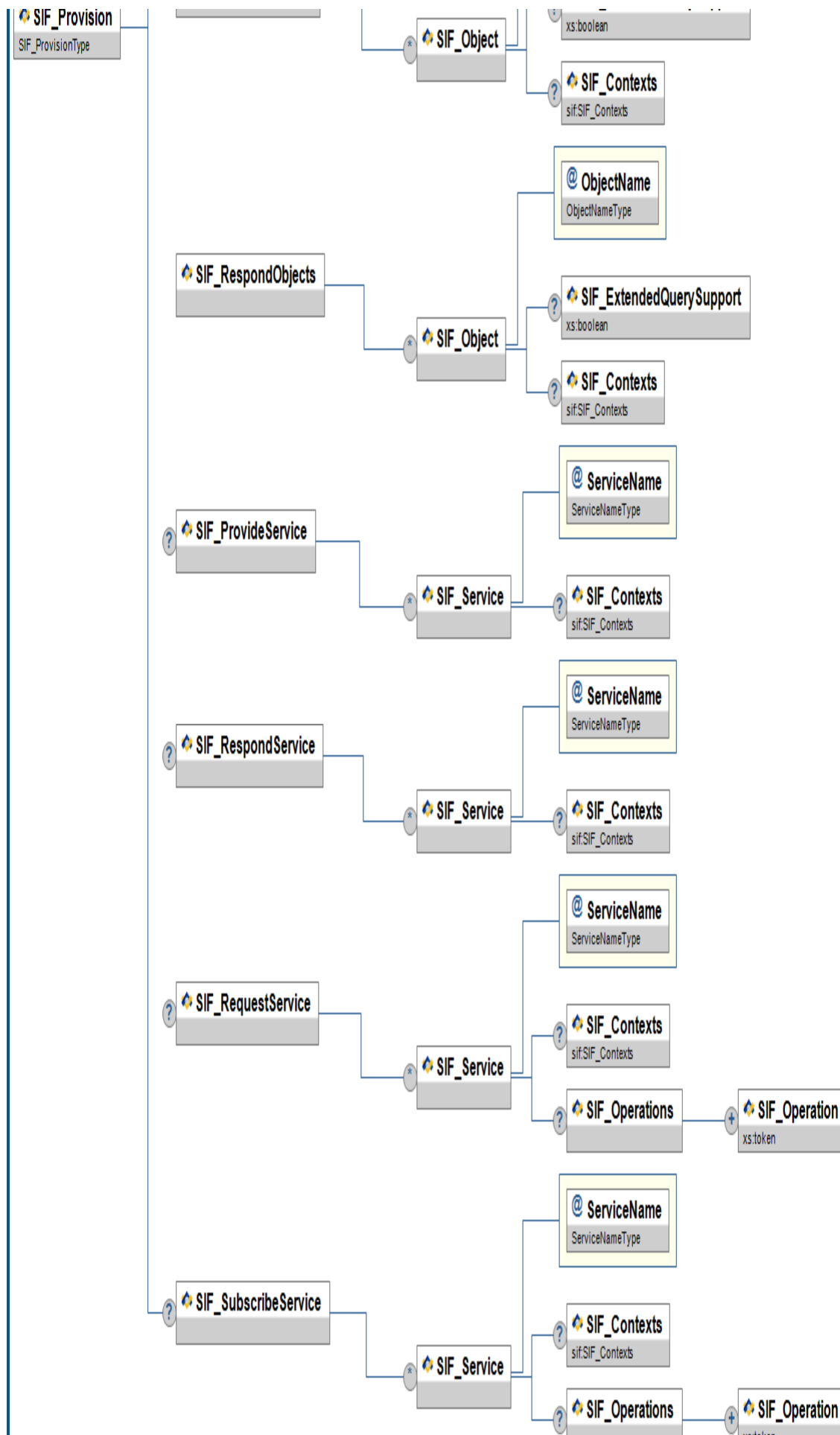


Figure 5.2.4-1: SIF_Provision

Element/@ Attribute	Char	Description	Type
SIF_Provision		Once registered, this message allows an agent to announce to the ZIS the functionality the agent will provide. The ZIS compares the functionality to its access control list and either returns a failure or a success. Upon success, the ZIS performs an atomic update of its provide/subscribe database entries for the agent to match the objects listed in this message and atomically updates other stored settings for the agent. A ZIS must not allow an agent to perform operations that it did not successfully announce. Agents should be aware that if the access control list changes after a successful <code>SIF_Provision</code> , some operations may still be rejected with access control errors.	
SIF_Header	M	Header information associated with this message.	<code>SIF_Header</code>
SIF_ProvideObjects	M	A list of objects an Agent wishes to provide.	
SIF_ProvideObjects/SIF_Object	OR		
@ ObjectName	M	The name of each object.	<code>ObjectNameType</code>
SIF_ProvideObjects/SIF_Object/ SIF_ExtendedQuerySupport	O	Whether or not <code>SIF_ExtendedQuery</code> is supported with regard to this object.	<code>xs:boolean</code>
SIF_ProvideObjects/SIF_Object/ SIF_Contexts	O	Applicable contexts for stated object support. If omitted, the context defaults to <code>SIF_Default</code> .	<code>SIF_Contexts</code>
SIF_SubscribeObjects	M	A list of objects to which an Agent wishes to subscribe.	
SIF_SubscribeObjects/SIF_Object	OR		
@ ObjectName	M	The name of each object.	<code>ObjectNameType</code>
SIF_SubscribeObjects/SIF_Object/ SIF_Contexts	O	Applicable contexts for stated object support. If omitted, the context defaults to <code>SIF_Default</code> .	<code>SIF_Contexts</code>
SIF_PublishAddObjects	M	A list of objects for which an Agent wishes to publish <code>Add</code> events.	
SIF_PublishAddObjects/SIF_Object	OR		

	Element/@Attribute	Char	Description	Type
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_PublishAddObjects/SIF_Object/ SIF_Contexts	O	Applicable contexts for stated object support. If omitted, the context defaults to <code>SIF_Default</code> .	SIF_Contexts
	SIF_PublishChangeObjects	M	A list of objects for which an Agent wishes to publish <code>Change</code> events.	
	SIF_PublishChangeObjects/ SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_PublishChangeObjects/ SIF_Object/SIF_Contexts	O	Applicable contexts for stated object support. If omitted, the context defaults to <code>SIF_Default</code> .	SIF_Contexts
	SIF_PublishDeleteObjects	M	A list of objects for which an Agent wishes to publish <code>Delete</code> events.	
	SIF_PublishDeleteObjects/ SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_PublishDeleteObjects/ SIF_Object/SIF_Contexts	O	Applicable contexts for stated object support. If omitted, the context defaults to <code>SIF_Default</code> .	SIF_Contexts
	SIF_RequestObjects	M	A list of objects an Agent wishes to request.	
	SIF_RequestObjects/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_RequestObjects/SIF_Object/ SIF_ExtendedQuerySupport	O	Optionally specify whether or not <code>SIF_ExtendedQuery</code> may be sent in requests for this object.	xs:boolean
	SIF_RequestObjects/SIF_Object/ SIF_Contexts	O	Applicable contexts for stated object support. If omitted, the context defaults to <code>SIF_Default</code> .	SIF_Contexts

	Element/@Attribute	Char	Description	Type
	SIF_RespondObjects	M	A list of objects for which an Agent wishes to handle requests, whether or not it is the Provider for each object.	
	SIF_RespondObjects/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_RespondObjects/SIF_Object/ SIF_ExtendedQuerySupport	O	Whether or not <code>SIF_ExtendedQuery</code> is supported with regard to this object.	xs:boolean
	SIF_RespondObjects/SIF_Object/ SIF_Contexts	O	Applicable contexts for stated object support. If omitted, the context defaults to <code>SIF_Default</code> .	SIF_Contexts
	SIF_ProvideService	O	A list of SIF Zone Services that the agent wishes to provide to the zone	
	SIF_ProvideService/SIF_Service	OR		
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	ServiceNameType
	SIF_ProvideService/SIF_Service/ SIF_Contexts	O	Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to <code>SIF_Default</code> .	SIF_Contexts
	SIF_RespondService	O	Indicates that the agent desires to respond to directed requests for one or more services in the SIF Zone	
	SIF_RespondService/SIF_Service	OR		
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	ServiceNameType
	SIF_RespondService/SIF_Service/ SIF_Contexts	O	Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to <code>SIF_Default</code> .	SIF_Contexts

	Element/@Attribute	Char	Description	Type
	SIF_RequestService	O	Indicates that the agent will make service calls to the specified SIF Zone Service by sending a SIF_ServiceInput message. This is an optional element that is used for allowing agents to report all of their expected activities within a zone. It is not used operationally within the zone, and failing to submit this element while provisioning will not prevent the agent from making service calls.	
	SIF_RequestService/SIF_Service	OR		
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	ServiceNameType
	SIF_RequestService/SIF_Service/ SIF_Contexts	O	Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to SIF_Default.	SIF_Contexts
	SIF_RequestService/SIF_Service/ SIF_Operations	O	The agent is not required to specify which operations it will invoke on the specified SIF Zone Service. However, the agent can do so if it desires report all of the activity it does within a zone.	List
	SIF_RequestService/SIF_Service/ SIF_Operations/SIF_Operation	MR	A specific method that the agent invokes the specified SIF Zone Service	xs:token
	SIF_SubscribeService	O	Indicates that the agent desires to subscribe to one or more events emitted by the specified service	
	SIF_SubscribeService/SIF_Service	OR		
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	ServiceNameType
	SIF_SubscribeService/SIF_Service/ SIF_Contexts	O	Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to SIF_Default.	SIF_Contexts
	SIF_SubscribeService/SIF_Service/ SIF_Operations	O	If SIF_Operations is not present, then the agent desires to subscribe to all events emitted by the service within the given context	List
	SIF_SubscribeService/SIF_Service/ SIF_Operations/SIF_Operation	MR	A specific event that the agent desires to subscribe to	xs:token

Table 5.2.4-1: SIF_Provision


```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Provision>
    <SIF_Header>
      <SIF_MsgId>A10F92EB649F4A648B5BFC44C7FD965C</SIF_MsgId>
      <SIF_Timestamp>2006-08-18T11:23:11-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_ProvideObjects>
      <SIF_Object ObjectName="StudentPersonal" />
      <SIF_Object ObjectName="StudentSchoolEnrollment" />
    </SIF_ProvideObjects>
    <SIF_SubscribeObjects>
      <SIF_Object ObjectName="StudentPicture" />
    </SIF_SubscribeObjects>
    <SIF_PublishAddObjects>
      <SIF_Object ObjectName="StudentPersonal" />
      <SIF_Object ObjectName="StudentSchoolEnrollment" />
    </SIF_PublishAddObjects>
    <SIF_PublishChangeObjects>
      <SIF_Object ObjectName="StudentPersonal" />
      <SIF_Object ObjectName="StudentSchoolEnrollment" />
    </SIF_PublishChangeObjects>
    <SIF_PublishDeleteObjects>
      <SIF_Object ObjectName="StudentPersonal" />
      <SIF_Object ObjectName="StudentSchoolEnrollment" />
    </SIF_PublishDeleteObjects>
    <SIF_RequestObjects>
      <SIF_Object ObjectName="StudentPicture" />
    </SIF_RequestObjects>
    <SIF_RespondObjects>
      <SIF_Object ObjectName="StudentPersonal" />
      <SIF_Object ObjectName="StudentSchoolEnrollment" />
    </SIF_RespondObjects>
  </SIF_Provision>
</SIF_Message>

```

Example 5.2.4-1: SIF_Provision

5.2.5 SIF_Register

SIF_Register is the message for registering an agent with a ZIS. An agent must be registered before it sends out other SIF messages. **SIF_Register** serves to provide the ZIS with the sender's identification information as well as to provide the information that the ZIS will need to contact this agent, should it register in Push mode.

Once a sender registers in the ZIS with the **SIF_Register** message, the sender can use the **SIF_SourceId** value in the header of all other outgoing messages as its identification. It is not necessary to send a **SIF_Register** message each time your agent starts up but it is not an error to do so. If there are any blocked events in the Agent's queue when a ZIS receives the **SIF_Register** message, the blocks will be removed.

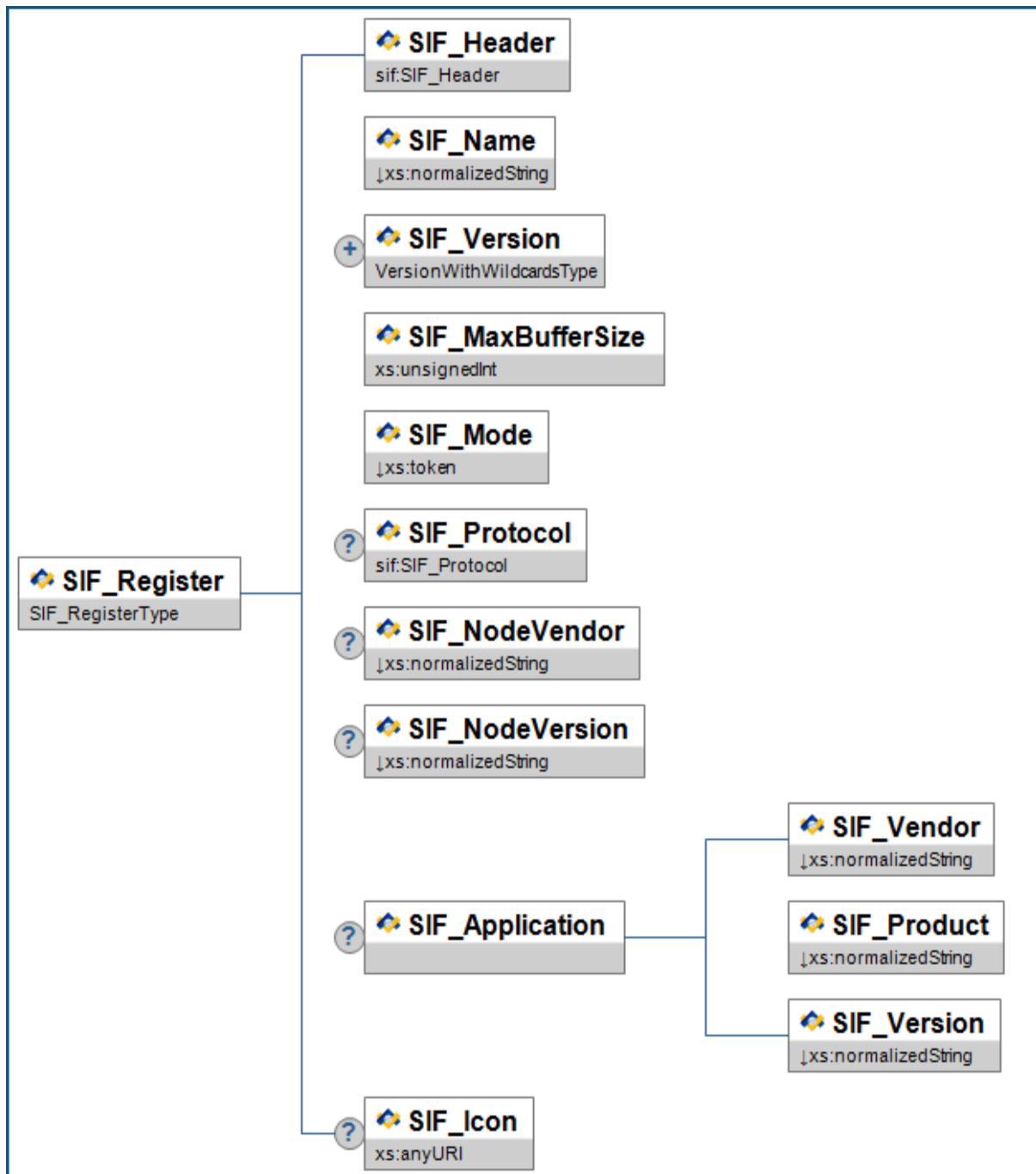


Figure 5.2.5-1: SIF_Register

Element/@Attribute	Char	Description	Type
SIF_Register	M	SIF_Register is the message for registering an agent with a ZIS.	
SIF_Header	M	Header information associated with this message.	SIF_Header
SIF_Name	M	This is the descriptive name of the agent that is registering (i.e. Ramsey Media Center).	xs:normalizedString <div>xs:maxLength 64</div>

Element/@Attribute	Char	Description	Type
SIF_Version	MR	<p>Specifies the SIF Implementation Specification version(s) defining messages the agent can receive. If the ZIS cannot communicate in this format, it should reject the request.</p> <p>The format of SIF_Version values can be found in Version Numbers. In a SIF_Register message, an individual SIF_Version element may also contain the following wildcards:</p> <p>* - Any SIF version</p> <p><major version>.* - Any minor version plus revisions within a major version (e.g., 1.*)</p> <p><major version>.<minor version><r>* - Any revision within a minor version (e.g., 1.1r*)</p> <p>Note: As wildcarding was first introduced in version 1.1 of the specification, 1.* does not match versions 1.0, 1.0r1 or 1.0r2. 1.1 or later agents that register with 1.* and wish to also receive messages from pre-1.1 agents must include SIF_Version element(s) with the supported pre-1.1 versions.</p>	VersionWithWildcardsType
SIF_MaxBufferSize	M	Specifies the maximum size of a packet to be returned by the ZIS. The ZIS may return packets smaller than, or equal to, the maximum value.	xs:unsignedInt
SIF_Mode	M	Specifies the communication mode (Pull or Push) as chosen by the message sender.	values: Push Pull
SIF_Protocol	C	If SIF_Mode is Push, SIF_Protocol contains protocol information for contacting the agent in Push mode. A Pull-mode agent does not need to send SIF_Protocol; if received, a ZIS ignores it.	SIF_Protocol
SIF_NodeVendor	O	The vendor of the SIF agent.	xs:normalizedString <div>xs:maxLength256</div>
SIF_NodeVersion	O	<p>The agent version number. The format of this field is undefined, but it should match the format used in the agent's conformance statement, if the agent is SIF Certified.</p> <p>Examples 2.0.1.11</p>	xs:normalizedString <div>xs:maxLength32</div>
SIF_Application	O	Contains information about the vendor of the product that the agent represents.	
SIF_Application/SIF_Vendor	M	The name of the company of the product that this agent supports.	xs:normalizedString <div>xs:maxLength256</div>

Element/@Attribute	Char	Description	Type
SIF_Application/SIF_Product	M	The name of the product that this agent supports.	xs:normalizedString xs:maxLength 256
SIF_Application/SIF_Version	M	The version of the product. This field is informative only.	xs:normalizedString xs:maxLength 32
SIF_Icon	O	HTTP URL referencing an icon for graphical representation of the application/agent. Should range from 16x16 pixels to 128x128 pixels and be of an <code>image</code> MIME type commonly supported by Web browsers (e.g. PNG, JPEG, GIF). Agents may optionally follow the more restrictive guidelines at [FAVICON] .	xs:anyURI

Table 5.2.5-1: SIF_Register

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Register>
    <SIF_Header>
      <SIF_MsgId>14BA09653261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-06:00</SIF_Timestamp>
      <SIF_SourceId>AcmeAgent</SIF_SourceId>
    </SIF_Header>
    <SIF_Name>Acme Agent for WAP 2.x</SIF_Name>
    <SIF_Version>2.5</SIF_Version>
    <SIF_MaxBufferSize>524288</SIF_MaxBufferSize>
    <SIF_Mode>Push</SIF_Mode>
    <SIF_Protocol Type="HTTPS" Secure="Yes">
      <SIF_URL>https://AcmeHost:8030/StudentAdmin</SIF_URL>
    </SIF_Protocol>
    <SIF_NodeVersion>2.0.1.20</SIF_NodeVersion>
    <SIF_Application>
      <SIF_Vendor>Acme Consulting</SIF_Vendor>
      <SIF_Product>Web Administration Portal 5.x</SIF_Product>
      <SIF_Version>5.1.2</SIF_Version>
    </SIF_Application>
  </SIF_Register>
</SIF_Message>

```

Example 5.2.5-1: SIF_Register

5.2.6 SIF_Request

This message is used to request information in SIF data objects from other SIF nodes. It optionally specifies the query criteria with which to match objects, as well as an optional subset of elements to be returned for matching objects.

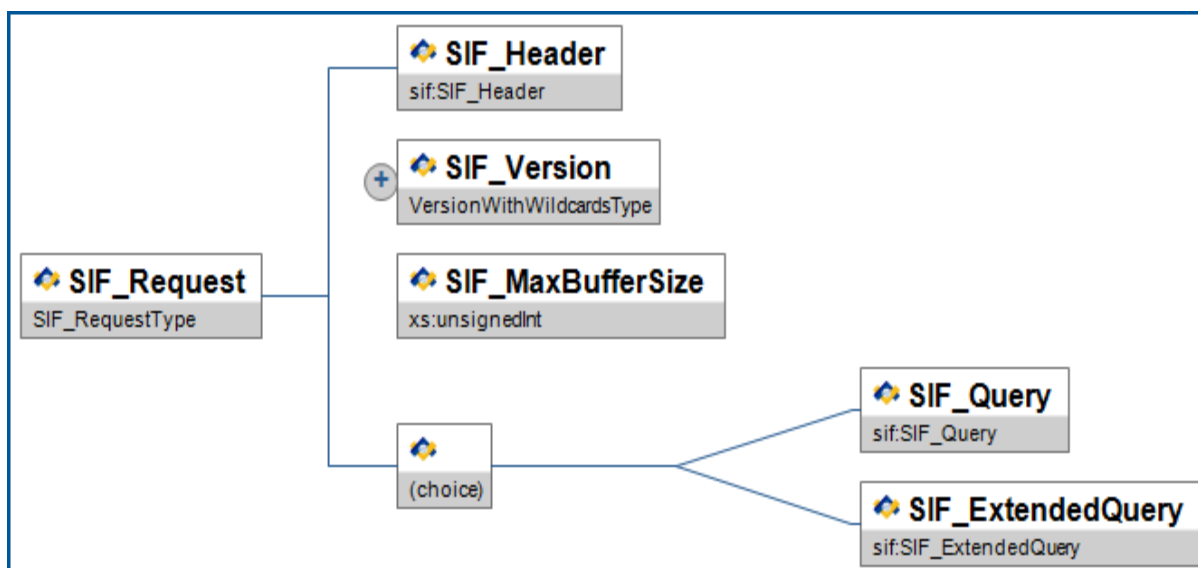


Figure 5.2.6-1: SIF_Request

Element/@ Attribute	Char	Description	Type
SIF_Request	M	SIF_Request is used to request information in SIF data objects from other SIF nodes.	
SIF_Header	M	Header information associated with this message.	SIF_Header
SIF_Version	MR	Specifies which SIF Implementation Specification version should be used when returning the response data; wildcards are allowed. The responding agent SHOULD return data using the highest version it supports that falls within the specified versions.	VersionWithWildcardsType
SIF_MaxBufferSize	M	Specifies the maximum size of a response packet to be returned to the requester. The responder may return packets smaller than, or equal to, the maximum value. To guarantee delivery of response packets, requesting agents must not specify a SIF_MaxBufferSize greater than its registered SIF_Register/SIF_MaxBufferSize .	xs:unsignedInt
SIF_Query	C	Either SIF_Query or SIF_ExtendedQuery must be specified, which contain the criteria to be used to match response objects.	SIF_Query
SIF_ExtendedQuery	C		SIF_ExtendedQuery

Table 5.2.6-1: SIF_Request

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Request>
    <SIF_Header>
      <SIF_MsgId>A3E90785EFDA330DACB00785EFDA330D</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_Version>2.*</SIF_Version>
    <SIF_MaxBufferSize>1048576</SIF_MaxBufferSize>
    <SIF_Query>
      <SIF_QueryObject ObjectName="LibraryPatronStatus" />
      <SIF_ConditionGroup Type="None">
        <SIF_Conditions Type="None">
          <SIF_Condition>
            <SIF_Element>@SIF_RefObject</SIF_Element>
            <SIF_Operator>EQ</SIF_Operator>
            <SIF_Value>StaffPersonal</SIF_Value>
          </SIF_Condition>
        </SIF_Conditions>
      </SIF_ConditionGroup>
    </SIF_Query>
  </SIF_Request>
</SIF_Message>

```

Example 5.2.6-1: SIF_Request

5.2.7 SIF_Response

SIF_Response is used to respond to a SIF_Request message. A response may span multiple SIF_Response messages.

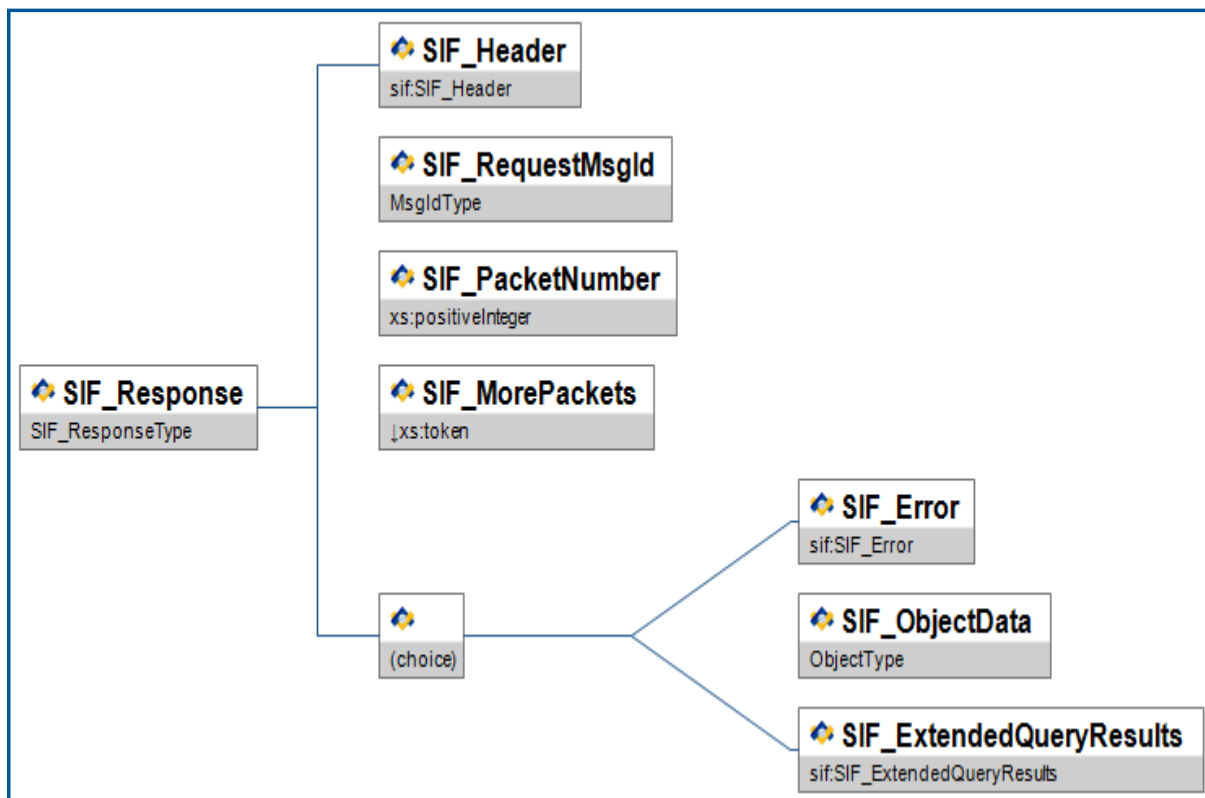


Figure 5.2.7-1: SIF_Response

Element/@ Attribute	Char	Description	Type
SIF_Response	M	SIF_Response is used to respond to a SIF_Request message.	

Element/@Attribute	Char	Description	Type
SIF_Header	M	Header information associated with this message. The <code>SIF_DestinationId</code> needs to be the <code>SIF_SourceId</code> of the original <code>SIF_Request</code> message being processed.	<code>SIF_Header</code>
SIF_RequestMsgId	M	This is the message Id of the <code>SIF_Request</code> message being processed. It provides a unique match between a <code>SIF_Response</code> and a previous <code>SIF_Request</code> . Since the Id of each message from an agent is unique, the receiver of a <code>SIF_Response</code> message will be able to relate the <code>SIF_Response</code> to a <code>SIF_Request</code> that it sent out previously.	<code>MsgIdType</code>
SIF_PacketNumber	M	<p>This element represents the index of the <code>SIF_Response</code> message in the sequence of packets that make up a complete response. Its value must be in the range of 1 through n, with n equal to the total number of packets that make up a response.</p> <p>The receiver of a <code>SIF_Response</code> message, with the help of the <code>SIF_MorePackets</code> and <code>SIF_PacketNumber</code> element in each incoming <code>SIF_Response</code> message, will be able to interpret and process each <code>SIF_Response</code> as part of a complete response to a previous <code>SIF_Request</code>.</p>	<code>xs:positiveInteger</code>
SIF_MorePackets	M	<p>This element provides an indication as to whether there are more packets besides this one to make up a complete response.</p> <p>The necessity of this element stems from the requirement on an agent to break response data to fit into the <code>SIF_MaxBufferSize</code> provided in the <code>SIF_Request</code>. Agents may also break response data into multiple packets for the benefit of improving performance or for circumventing limitations of the underlying network infrastructure.</p> <p>When this element's value is equal to <code>No</code>, it is an indication from the sender to the receiver that it has already sent out all the packets that make up a complete response for a <code>SIF_Request</code> as indicated by the <code>SIF_RequestMsgId</code> element.</p>	values: <code>Yes</code> <code>No</code>
SIF_Error	C	<p>The responder returns <code>SIF_Error</code>, <code>SIF_ObjectData</code> or <code>SIF_ExtendedQueryResults</code>.</p> <p>This element allows the Responder to report an error condition that occurs while processing the <code>SIF_Request</code>.</p> <p>If a <code>SIF_Error</code> element is present, the requesting agent must not expect to receive further <code>SIF_Responses</code> to the <code>SIF_Request</code>.</p>	<code>SIF_Error</code>
SIF_ObjectData	C	The <code>SIF_ObjectData</code> element contains the data objects matching the supplied criteria in the <code>SIF_Request</code> message if the <code>SIF_Request</code> contained <code>SIF_Query</code> . If the <code>SIF_Request</code> contained <code>SIF_ExtendedQuery</code> , include <code>SIF_ExtendedQueryResults</code> .	<code>ObjectType</code>

Element/@Attribute	Char	Description	Type
SIF_ExtendedQueryResults	C	This element contains the elements requested by SIF_ExtendedQuery in SIF_Request.	SIF_ExtendedQueryResults

Table 5.2.7-1: SIF_Response

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Response>
    <SIF_Header>
      <SIF_MsgId>1BCD10580EF250789012AC0554321EA2</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>SISAgent</SIF_SourceId>
      <SIF_DestinationId>NetworkAgent</SIF_DestinationId>
    </SIF_Header>
    <SIF_RequestMsgId>FE1078BA3261545A319059376B3A4898</SIF_RequestMsgId>
    <SIF_PacketNumber>1</SIF_PacketNumber>
    <SIF_MorePackets>No</SIF_MorePackets>
    <SIF_ObjectData>
      <StudentPersonal RefId="E3E34B359D75101A8C3D00AA00184753">
        <Name Type="04">
          <LastName>Johnson</LastName>
          <FirstName>Alicia</FirstName>
        </Name>
      </StudentPersonal>
    </SIF_ObjectData>
  </SIF_Response>
</SIF_Message>

```

Example 5.2.7-1: Sample single-packet SIF_Response to a SIF_Request for the Name element from a StudentPersonal object

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Response>
    <SIF_Header>
      <SIF_MsgId>322925BC9818433E8090D5110EE61DA3</SIF_MsgId>
      <SIF_Timestamp>2006-04-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>SISAgent</SIF_SourceId>
      <SIF_DestinationId>NetworkAgent</SIF_DestinationId>
    </SIF_Header>
    <SIF_RequestMsgId>FE1078BA3261545A31905937B265CE01</SIF_RequestMsgId>
    <SIF_PacketNumber>1</SIF_PacketNumber>
    <SIF_MorePackets>Yes</SIF_MorePackets>
    <SIF_ObjectData>
      <StudentPersonal RefId="E3E34B359D75101A8C3D00AA00184753">
        <Name Type="04">
          <LastName>Johnson</LastName>
          <FirstName>Alicia</FirstName>
        </Name>
      </StudentPersonal>
    </SIF_ObjectData>
  </SIF_Response>
</SIF_Message>

```

Example 5.2.7-3: SIF_Response (first packet)


```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Response>
    <SIF_Header>
      <SIF_MsgId>322925BC9818433E8090D51256786BC9</SIF_MsgId>
      <SIF_Timestamp>2006-04-18T08:39:49-08:00</SIF_Timestamp>
      <SIF_SourceId>SISAgent</SIF_SourceId>
      <SIF_DestinationId>NetworkAgent</SIF_DestinationId>
    </SIF_Header>
    <SIF_RequestMsgId>FE1078BA3261545A31905937B265CE01</SIF_RequestMsgId>
    <SIF_PacketNumber>2</SIF_PacketNumber>
    <SIF_MorePackets>No</SIF_MorePackets>
    <SIF_ObjectData>
      <StudentPersonal RefId="F14B5B359D75101A8C3D00AA00184753">
        <Name Type="04">
          <LastName>Smith</LastName>
          <FirstName>Alicia</FirstName>
        </Name>
      </StudentPersonal>
    </SIF_ObjectData>
  </SIF_Response>
</SIF_Message>

```

Example 5.2.7-5: SIF_Response (second packet)

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Response>
    <SIF_Header>
      <SIF_MsgId>F557D40A1367455E9F01DED76E29260C</SIF_MsgId>
      <SIF_Timestamp>2006-04-18T08:43:08-08:00</SIF_Timestamp>
      <SIF_SourceId>SISAgent</SIF_SourceId>
      <SIF_DestinationId>NetworkAgent</SIF_DestinationId>
    </SIF_Header>
    <SIF_RequestMsgId>971D7C7EF2684C7081A7765BF89FAD14</SIF_RequestMsgId>
    <SIF_PacketNumber>1</SIF_PacketNumber>
    <SIF_MorePackets>No</SIF_MorePackets>
    <SIF_ObjectData />
  </SIF_Response>
</SIF_Message>

```

Example 5.2.7-7: SIF_Response with no matching objects

5.2.8 SIF_Subscribe

This message is used to subscribe to event objects that are contained in this message.

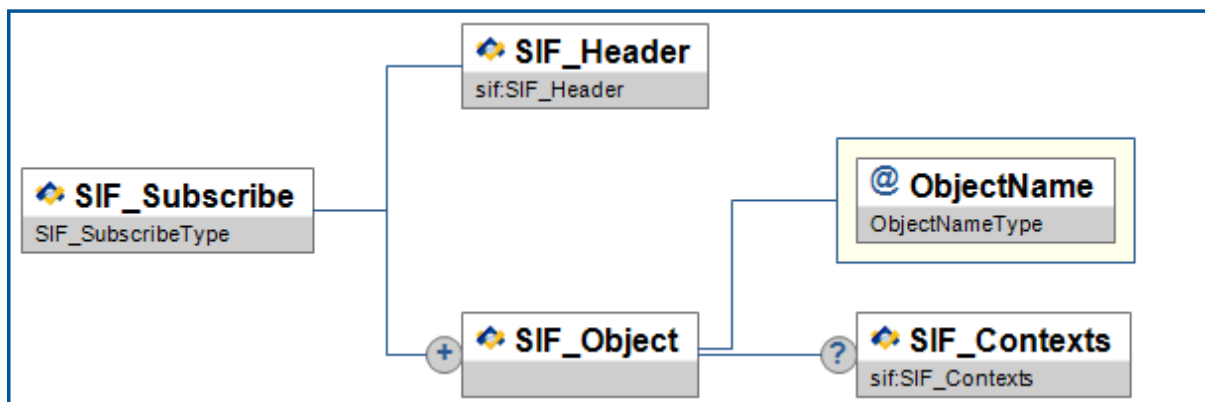


Figure 5.2.8-1: SIF_Subscribe

Element/@Attribute	Char	Description	Type
SIF_Subscribe	M	This message is used to subscribe to event objects that are contained in this message.	
SIF_Header	M	Header information associated with this message.	SIF_Header
SIF_Object	MR		
@ ObjectName	M	The name of the object that is being subscribed to. All valid SIF_Events for this object will be routed to the subscriber.	ObjectNameType
SIF_Object/SIF_Contexts	O	The contexts to which the subscription applies; if omitted, the context is SIF_Default.	SIF_Contexts

Table 5.2.8-1: SIF_Subscribe

```

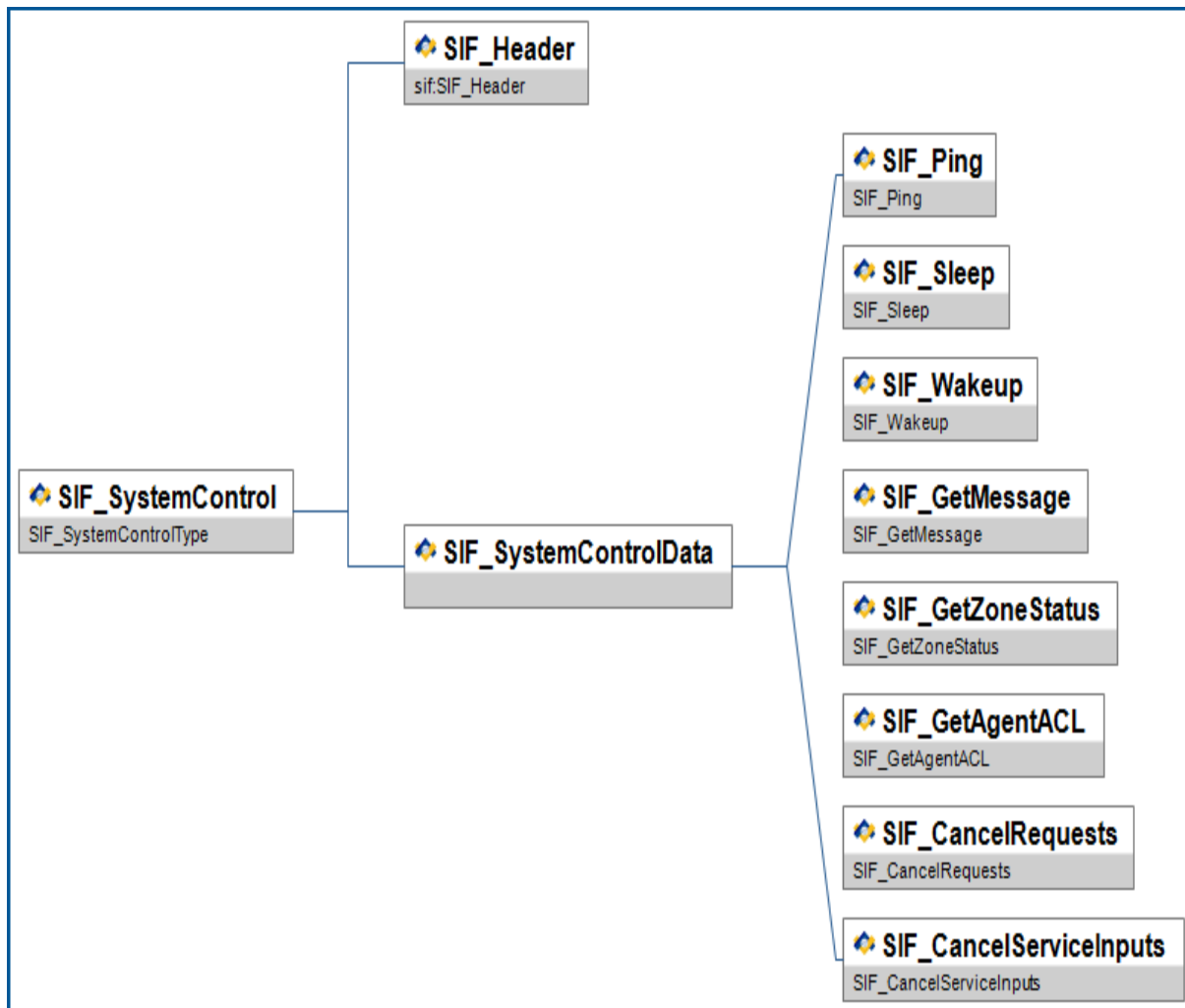
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Subscribe>
    <SIF_Header>
      <SIF_MsgId>AB2065FD3261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_Object ObjectName="StudentPersonal" />
    <SIF_Object ObjectName="StaffPersonal" />
  </SIF_Subscribe>
</SIF_Message>

```

Example 5.2.8-1: SIF_Subscribe

5.2.9 SIF_SystemControl

A SIF_SystemControl message is designed to control the flow of data between an agent and ZIS or vice-versa, and to synchronously retrieve data available from the ZIS. The SIF_SystemControl message is a container for a number of specialized control messages. SIF_SystemControl messages are handled immediately by receivers and are not persisted in a message queue for later delivery.

Figure 5.2.9-1: *SIF_SystemControl*

Element/@Attribute	Char	Description	Type
SIF_SystemControl	M	This message is designed to control the flow of data an agent and ZIS or vice-versa, and to synchronously retrieve data available from the ZIS.	
SIF_Header	M	Header information associated with this message.	SIF_Header
SIF_SystemControlData	M	This element holds the sub-message being sent.	choice of: SIF_Ping SIF_Sleep SIF_Wakeup SIF_GetMessage SIF_GetZoneStatus SIF_GetAgentACL SIF_CancelRequests SIF_CancelServiceInputs

Table 5.2.9-1: *SIF_SystemControl*

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>C332B8A9DFA5480AB89B6B6F62BE57B3</SIF_MsgId>
      <SIF_Timestamp>2006-12-27T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_Ping />
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>
```

Example 5.2.9-1: SIF_SystemControl

5.2.10 SIF_Ping

SIF_Ping is sent to detect if a ZIS or push-mode agent is ready to receive and process messages.



Figure 5.2.10-1: SIF_Ping

Element/@Attribute	Char	Description	Type
SIF_Ping	M	This sub-message detects if an a Push-Mode Agent or ZIS is ready to receive and process messages.	EMPTY

Table 5.2.10-1: SIF_Ping

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>C332B8A9DFA5480AB89B6B6F62BE57B3</SIF_MsgId>
      <SIF_Timestamp>2006-12-27T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_Ping />
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>
```

Example 5.2.10-1: SIF_Ping

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>AE9E2BD747B94F4D8545E41F482854C8</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T14:23:20-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseyZIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>9812ABFD3261545A31905937B265CE01</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>1</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example 5.2.10-3: SIF_SystemControl—SIF_Ping ("Okay" status)

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>3C11DFF1451C4E9A8A1F07E03C1D7FBB</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T14:24:31-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseyZIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>9812ABFD3261545A31905937B265CE01</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>8</SIF_Code>
      <SIF_Desc>Receiver is sleeping</SIF_Desc>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example 5.2.10-5: SIF_SystemControl—SIF_Ping ("Receiver is sleeping" status)

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>1594A3B29DD34786B5EA77998899F49F</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T14:24:31-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>9812ABFD3261545A31905937B265CE01</SIF_OriginalMsgId>
    <SIF_Error>
      <SIF_Category>10</SIF_Category>
      <SIF_Code>4</SIF_Code>
      <SIF_Desc>Unable to establish connection</SIF_Desc>
      <SIF_ExtendedDesc>Error 10061: Connection refused</SIF_ExtendedDesc>
    </SIF_Error>
  </SIF_Ack>
</SIF_Message>
```

Example 5.2.10-7: SIF_SystemControl—SIF_Ping (Transport error returned)

5.2.11 SIF_Sleep

The `SIF_Sleep` message allows an agent to notify a ZIS or a ZIS to notify a push-mode agent that it must not send any more messages to the sender of the `SIF_Sleep`. After the sender receives a `SIF_Ack` indicating that the message was received, the receiver must not send any further messages to the sender.

This message provides the ability to signal an agent or ZIS that the sender will be unable to process further messages until some time in the future. Reasons for sending a `SIF_Sleep` message include the sender is unable to process more data because of limited resources (i.e. disk storage, network bandwidth, etc.), or the sender is being temporarily shutdown and will be unable to receive messages.

Since the sender may send a `SIF_Sleep` message for a variety of reasons, if the receiver sends messages after a `SIF_Sleep` message but prior to receiving a `SIF_Wakeup` or `SIF_Register` message from the sender, an error must be returned. A transport error will occur or be returned if a connection cannot be established with the sender, or the sender may choose to receive the connection but return an error.

If the sender is an agent, the ZIS will continue to hold any messages for the agent in the queue but the ZIS will not send those messages until a `SIF_Wakeup` (or `SIF_Register`) message is received. If an agent is processing a message requiring additional `SIF_RequestS` to be sent to the ZIS and a `SIF_Sleep` message is received from the ZIS, the agent will not be able to retrieve the additional data. The agent must abort the processing of the message and only attempt to process the message after receiving a `SIF_Wakeup` message from the ZIS.

An agent or ZIS is not required to be able to send `SIF_Sleep` messages. However, if an agent or ZIS has the ability to send a `SIF_Sleep`, it must also be able to send a `SIF_Wakeup`. Although the sending of `SIF_Sleep` is optional, an agent or ZIS must always be able to process and respond appropriately to these messages if received.



Figure 5.2.11-1: SIF_Sleep

Element/@Attribute	Char	Description	Type
SIF_Sleep	M	This sub-message tells a receiver not to send any more messages to the sender.	EMPTY

Table 5.2.11-1: SIF_Sleep

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>1594A3B29DD34786B5EA77998899F49F</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T14:28:19-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_Sleep />
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>
```

Example 5.2.11-1: SIF_Sleep

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>9F5167FA5CA848F99EB27544B314AF4D</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T14:29:09-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseyZIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>1594A3B29DD34786B5EA77998899F49F</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>1</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example 5.2.11-3: SIF_Ack with "Okay" status in response to SIF_Sleep

5.2.12 SIF_Wakeup

When the "sleeping" agent or ZIS is ready to resume message processing, it will send a SIF_Wakeup message. This will signal the receiver that the sender is now able to process messages. Sending a SIF_Wakeup message without a previous SIF_Sleep message is permissible and is not considered an error.

If there are any blocked events in the Agent's queue when a ZIS receives the SIF_Wakeup message, the blocks will be removed.

Since a ZIS may choose to stop sending messages to an agent if a connection cannot be made with that agent, it is recommended that an agent send a SIF_Wakeup message to the ZIS upon agent startup.

An agent or ZIS is not required to be able to send SIF_Wakeup messages. Although the sending of SIF_Wakeup is optional, an agent or ZIS must always be able to process and respond to these messages correctly if received.



Figure 5.2.12-1: SIF_Wakeup

Element/@Attribute	Char	Description	Type
SIF_Wakeup	M	This message tells a receiver that the sender is able to process messages.	EMPTY

Table 5.2.12-1: SIF_Wakeup

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>715A32E026B0495A826DF84E821949BD</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T15:34:22-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_Wakeup />
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>
```

Example 5.2.12-1: SIF_Wakeup

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>5631E28868E3482EAA51B6CDE4145957</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T15:34:48-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseyZIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>715A32E026B0495A826DF84E821949BD</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>1</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example 5.2.12-3: SIF_Ack with an "Okay" status in response to SIF_Wakeup

5.2.12.1 SIF_Sleep/SIF_Wakeup versus SIF_Register/SIF_Unregister

Using the SIF_Wakeup message is the preferred method of communicating that an agent or ZIS is ready to process messages. This is preferable over the use of a SIF_Register message because a SIF_Register message specifies protocol information while the SIF_Sleep/SIF_Wakeup pair communicates flow control information. However, when a SIF_Register message is processed, the receiver must behave like a SIF_Wakeup message was also received.

It is important to note that while SIF_Sleep and SIF_Wakeup are opposites of one another, this is not the case with SIF_Register and SIF_Unregister. This is because a SIF_Unregister command removes essential agent configuration information such as the provision and subscription lists, which will not be specified by a subsequent SIF_Register command. (SIF_Unregister also causes any messages pending delivery to the agent to be purged from the agent's queue.) In other words, a SIF_Register alone will not reverse the effects of a SIF_Unregister.

5.2.13 SIF_GetMessage

The SIF_GetMessage message provides the mechanism for an agent to pull message from a ZIS. An agent sends a SIF_GetMessage and the ZIS returns the next available message, subject to Selective Message Blocking, wrapped in a SIF_Ack with a SIF_Status/SIF_Code of 0 and the message in the SIF_Status/SIF_Data element. If there are no messages to be returned, the ZIS returns a value of 9 in SIF_Status/SIF_Code.

If an agent is not registered with a mode of Pull the ZIS will return a SIF_Ack with an error category of Registration and an error code indicating the agent is registered in Push mode.

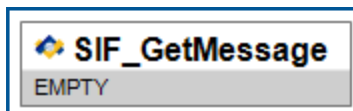


Figure 5.2.13-1: SIF_GetMessage

Element/@Attribute	Char	Description	Type
SIF_GetMessage	M	This message tells the ZIS to return the first available message to the agent, subject to Selective Message Blocking.	EMPTY

Table 5.2.13-1: SIF_GetMessage

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>B0E80A74265A4A75ADDC0ECC50AEF737</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T15:54:32-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_GetMessage />
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>

```

Example 5.2.13-1: SIF_GetMessage

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>9861A45AAC364607938A7DB440514DDF</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T15:54:42-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>B0E80A74265A4A75ADDC0ECC50AEF737</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>0</SIF_Code>
    <SIF_Data>
      <SIF_Message Version="2.5">
        <SIF_Event>
          <SIF_Header>
            <SIF_MsgId>AB34DC093261545A31905937B265CE01</SIF_MsgId>
            <SIF_Timestamp>2006-10-14T15:40:12-08:00</SIF_Timestamp>
            <SIF_SourceId>RamseySIS</SIF_SourceId>
          </SIF_Header>
          <SIF_ObjectData>
            <SIF_EventObject ObjectName="StudentPersonal" Action="Change">
              <StudentPersonal RefId="D3E34B359D75101A8C3D00AA001A1652">
                <PhoneNumberList>
                  <PhoneNumber Type="0096">
                    <Number>(312) 555-1234</Number>
                  </PhoneNumber>
                </PhoneNumberList>
              </StudentPersonal>
            </SIF_EventObject>
          </SIF_ObjectData>
        </SIF_Event>
      </SIF_Message>
    </SIF_Data>
  </SIF_Status>
</SIF_Ack>
</SIF_Message>

```

Example 5.2.13-3: SIF_Ack in response to SIF_GetMessage


```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>9861A45AAC364607938A7DB440514DDF</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T15:54:42-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>B0E80A74265A4A75ADDC0ECC50AEF737</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>9</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example 5.2.13-5: SIF_Ack in response to SIF_GetMessage (no message in queue)

5.2.14 SIF_GetZoneStatus

The SIF_GetZoneStatus message provides the agent with the ability to synchronously retrieve the current status of the zone, by-passing the asynchronous nature of retrieving the zone status by sending a SIF_Request for SIF_ZoneStatus and waiting for the arrival of the SIF_ZoneStatus response at the top of its queue. Agents may also use the asynchronous model for requesting SIF_ZoneStatus, if and when desired.

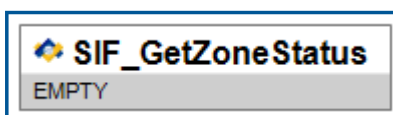


Figure 5.2.14-1: SIF_GetZoneStatus

Element/@ Attribute	Char	Description	Type
SIF_GetZoneStatus	M	This message tells the ZIS to return the current SIF_ZoneStatus in a SIF_Ack.	EMPTY

Table 5.2.14-1: SIF_GetZoneStatus

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>91401B5073F54AB1AEB63E51764C77A</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T16:09:54-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_GetZoneStatus />
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>
```

Example 5.2.14-1: SIF_GetZoneStatus

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>C4BF5F868EEC4A41AF7DAF316C4E89DC</SIF_MsgId>
      <SIF_Timestamp>2006-10-14T16:10:42-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyZIS</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>91401B5073F54AB1AEBC63E51764C77A</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>0</SIF_Code>
      <SIF_Data>
        <SIF_ZoneStatus ZoneId="SIFExampleZone">...</SIF_ZoneStatus>
      </SIF_Data>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>

```

Example 5.2.14-3: SIF_Ack containing SIF_ZoneStatus

5.2.15 SIF_GetAgentACL

The SIF_GetAgentACL message provides the agent with the ability to synchronously retrieve its Access Control List permissions in the Zone via SIF_AgentACL. Agents may also use the asynchronous model of SIF_Request for requesting SIF_AgentACL, if and when desired.

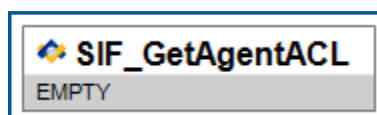


Figure 5.2.15-1: SIF_GetAgentACL

Element/@Attribute	Char	Description	Type
SIF_GetAgentACL	M	This message tells the ZIS to return the Agent's ACL permissions in a SIF_Ack.	EMPTY

Table 5.2.15-1: SIF_GetAgentACL

5.2.16 SIF_CancelRequests

The SIF_SystemControl—SIF_CancelRequests message allows an Agent or ZIS to notify a ZIS or Push-Mode Agent, respectively, that the specified SIF_Requests should be cancelled, whether pending or in process. Handling by a Push-Mode Agent is optional; if unsupported, the Agent returns a Generic Message Handling error upon receipt of the SIF_SystemControl message, error code "Message not supported."

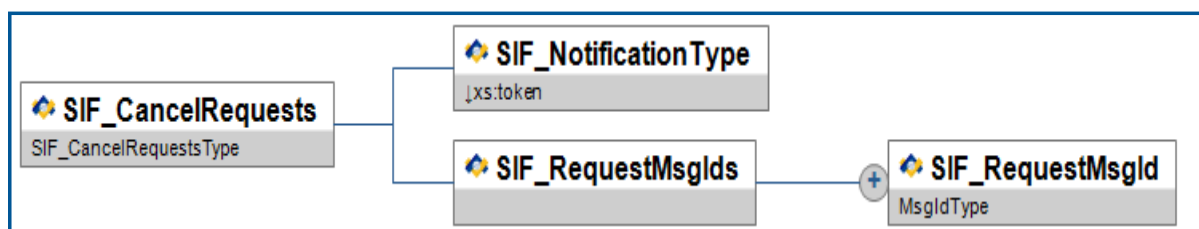


Figure 5.2.16-1: SIF_CancelRequests

Element/@Attribute	Char	Description	Type
SIF_CancelRequests	M	This sub-message asks a receiver (ZIS or Push-Mode Agent) to cancel the specified <code>SIF_RequestS</code> , pending or in process.	
SIF_NotificationType	M		values: Standard ZIS will send a "final" <code>SIF_Response</code> for each cancelled <code>SIF_Request</code> . None No further <code>SIF_Respones</code> for these requests will be placed in the Agent's queue.
SIF_RequestMsgIds	M	The list of <code>SIF_RequestS</code> to cancel.	List
SIF_RequestMsgIds/SIF_RequestMsgId	MR	This is the <code>SIF_MsgId</code> of the <code>SIF_Request</code> message being cancelled.	MsgIdType

Table 5.2.16-1: SIF_CancelRequests

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>C332B8A9DFA5480AB89B6B6F62BE57B3</SIF_MsgId>
      <SIF_Timestamp>2006-12-27T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>AcmeAgent</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_CancelRequests>
        <SIF_NotificationType>None</SIF_NotificationType>
        <SIF_RequestMsgIds>
          <SIF_RequestMsgId>C332B8A9DFA5480AB89B6B6F62BE57B3</SIF_RequestMsgId>
          <SIF_RequestMsgId>1058ABCDE028D076F08365109BE7C892</SIF_RequestMsgId>
        </SIF_RequestMsgIds>
      </SIF_CancelRequests>
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>

```

Example 5.2.16-1: SIF_CancelRequests

5.2.17 SIF_CancelServiceInputs

The `SIF_SystemControl` message is already part of the SIF Infrastructure. This messages allows for synchronous communication between an agent and a ZIS. The `SIF_SystemControl - SIF_CancelServiceInputs` allows an agent or ZIS to notify the other party that the specified `SIF_ServiceInput` should be cancelled.

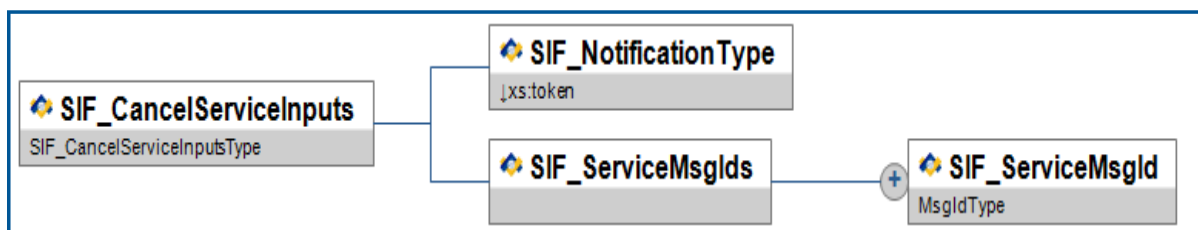


Figure 5.2.17-1: SIF_CancelServiceInputs

Element/@Attribute	Char	Description	Type
SIF_CancelServiceInputs	M	The SIF_SystemControl message is already part of the SIF Infrastructure. This messages allows for synchronous communication between an agent and a ZIS. The SIF_SystemControl - SIF_CancelServiceInputs allows an agent or ZIS to notify the other party that the specified SIF_ServiceInput should be cancelled.	
SIF_NotificationType	M		values: Standard ZIS will send a "final" SIF_ServiceOutput for each cancelled SIF_ServiceInput. None No further SIF_ServiceOutputs for these requests will be placed in the Agent's queue.
SIF_ServiceMsgIds	M	The list of SIF_ServiceInputs to cancel.	List
SIF_ServiceMsgIds/SIF_ServiceMsgId	MR	This is the SIF_ServiceMsgId of the SIF_ServiceInput message being cancelled.	MsgIdType

Table 5.2.17-1: SIF_CancelServiceInputs

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_SystemControl>
    <SIF_Header>
      <SIF_MsgId>C332B8A9DFA5480AB89B6B6F62BE57B3</SIF_MsgId>
      <SIF_Timestamp>2006-12-27T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>AcmeAgent</SIF_SourceId>
    </SIF_Header>
    <SIF_SystemControlData>
      <SIF_CancelServiceInputs>
        <SIF_NotificationType>None</SIF_NotificationType>
        <SIF_ServiceMsgIds>
          <SIF_ServiceMsgId>C332B8A9DFA5480AB89B6B6F62BE57B3</SIF_ServiceMsgId>
          <SIF_ServiceMsgId>1058ABCDE028D076F08365109BE7C892</SIF_ServiceMsgId>
        </SIF_ServiceMsgIds>
      </SIF_CancelServiceInputs>
    </SIF_SystemControlData>
  </SIF_SystemControl>
</SIF_Message>

```

Example 5.2.17-1: SIF_CancelServiceInputs

5.2.18 SIF_Unprovide

This message performs the opposite function of SIF_Provide. It removes the message sender as a provider of the data objects contained in this message.

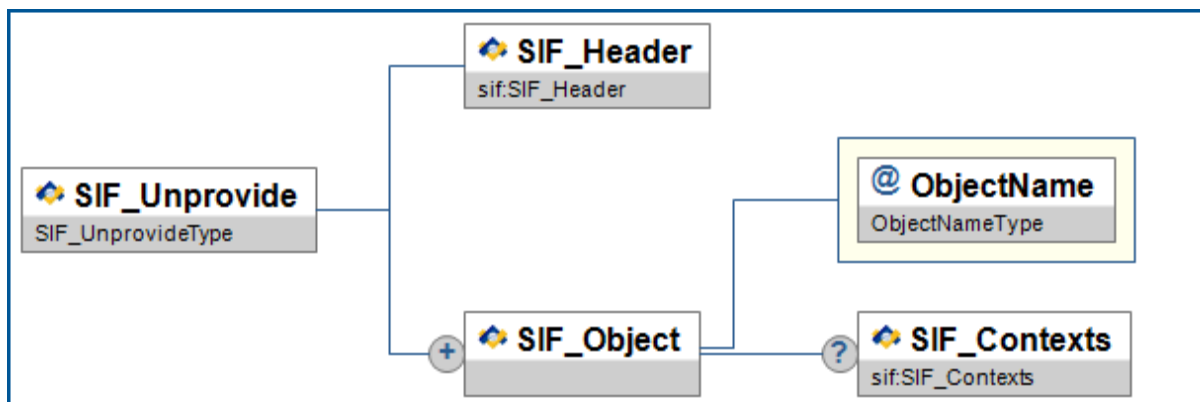


Figure 5.2.18-1: SIF_Unprovide

Element/@Attribute	Char	Description	Type
SIF_Unprovide	M	This message performs the opposite function of SIF_Provide.	
SIF_Header	M	Header information associated with this message.	SIF_Header
SIF_Object	MR	This is the object that is being removed from the provider list.	
@ ObjectName	M	The name of the object that is being removed.	ObjectNameType
SIF_Object/SIF_Contexts	O	The contexts from which the object is being removed; if omitted, the context is SIF_Default.	SIF_Contexts

Table 5.2.18-1: SIF_Unprovide

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Unprovide>
    <SIF_Header>
      <SIF_MsgId>76EFAB543261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseySIS</SIF_SourceId>
    </SIF_Header>
    <SIF_Object ObjectName="StudentPersonal" />
    <SIF_Object ObjectName="StaffPersonal" />
  </SIF_Unprovide>
</SIF_Message>

```

Example 5.2.18-1: SIF_Unprovide

5.2.19 SIF_Unregister

This message allows an agent to remove any association it has with the ZIS. By sending this message, the ZIS will remove all provisions and subscriptions it maintains for the sender and discards any messages pending for the agent.



Figure 5.2.19-1: SIF_Unregister

Element/@Attribute	Char	Description	Type
SIF_Unregister	M	This message is used to unregister an agent from a Zone.	
SIF_Header	M	Header information contained in the message.	SIF_Header

Table 5.2.19-1: SIF_Unregister

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Unregister>
    <SIF_Header>
      <SIF_MsgId>1057FABD3261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyFOOD</SIF_SourceId>
    </SIF_Header>
  </SIF_Unregister>
</SIF_Message>

```

Example 5.2.19-1: SIF_Unregister

5.2.20 SIF_Unsubscribe

This message performs the opposite function of SIF_Subscribe. It removes the message sender as a subscriber to the SIF_Events contained in this message.

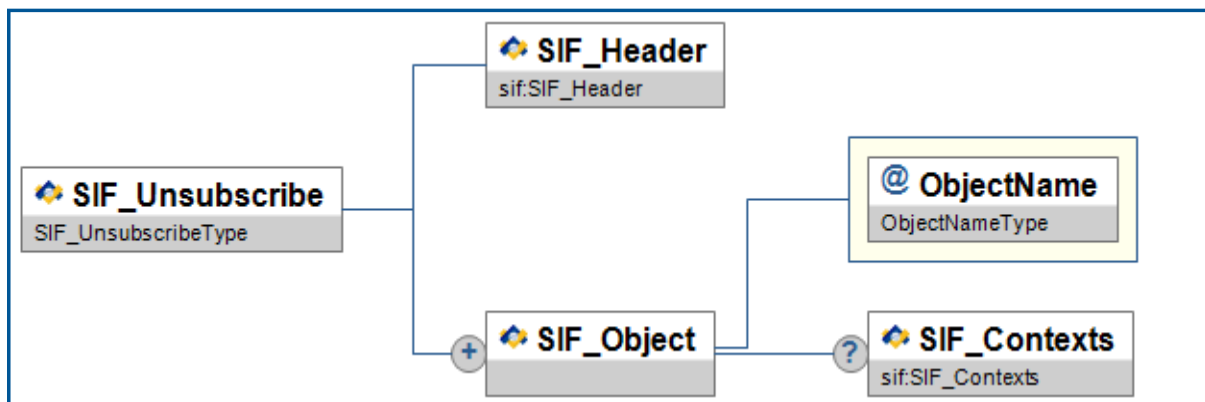


Figure 5.2.20-1: SIF_Unsubscribe

Element/@Attribute	Char	Description	Type
SIF_Unsubscribe	M	This message is used to unsubscribe from SIF_EventS.	
SIF_Header	M	Header information associated with this message.	SIF_Header

	Element/@Attribute	Char	Description	Type
	SIF_Object	MR		
@	ObjectName	M	The name of the SIF object from which the agent should be unsubscribed. Events pertaining to this object published after successful unsubscription will no longer be queued for delivery to the agent. Events already queued for delivery to the agent prior to unsubscription will be delivered.	ObjectNameType
	SIF_Object/SIF_Contexts	O	The applicable contexts; if omitted, the context is SIF_Default.	SIF_Contexts

Table 5.2.20-1: SIF_Unsubscribe

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Unsubscribe>
    <SIF_Header>
      <SIF_MsgId>101AE3703261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T20:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyFOOD</SIF_SourceId>
    </SIF_Header>
    <SIF_Object ObjectName="StudentPersonal" />
    <SIF_Object ObjectName="StaffPersonal" />
  </SIF_Unsubscribe>
</SIF_Message>

```

Example 5.2.20-1: SIF_Unsubscribe

5.2.21 SIF_ServiceInput

This message is used to invoke a method that is exposed by a SIF Zone Service.

The SIF Zone Service specification may state that multiple SIF_ServiceInput messages may be sent to comprise a single invocation of a zone service method. If this is the case, the must be packetized using the following rules.

When an agent is creating SIF_ServiceInput packets, it **MUST** attempt to ensure that each packet is no larger than the maximum SIF_MaxBufferSize supported by the service. The default SIF_MaxBufferSize for any SIF Zone service is 64k. Each SIF_ServiceInput message be packetized using a maximum 64K to a SIF Zone Service by a service client (agent). A higher buffer size **MAY** be used only if the service client (agent) has verified that the service can accept a higher buffer size.

The size of a SIF_ServiceInput message **SHOULD** be less than or equal 65,536 bytes, unless one of the following is true:

1. The SIF Zone Service is defined as requiring a larger buffer size for unsolicited messages. If this is the case, the SIF Specification for the SIF Zone service specifies the SIF_MaxBufferSize that is defined for this service. If that is the case, then the size specified in the SIF Zone Service definition becomes the maximum allowable buffer size.
2. The service client has determined that a higher buffer size is supported by the publisher of the service, either through static configuration by the zone administrator or dynamically at runtime by examination of SIF_ZoneStatus.

When a service client is sending multiple SIF_ServiceInput packets, the following rules apply:

1. If a single subsequent packet does not fit within the define SIF_MaxBufferSize, the agent **MUST**, in addition to acknowledging receipt of the message to the ZIS, send a SIF_ServiceInput message to the service with the SIF_Error element populated to indicate the nature of the error, and the SIF_MorePackets element set to indicate that no further packets will be sent in for the SIF_ServiceInput.
2. If the service client encounters an error and is unable to continue sending SIF_ServiceInput packets, the service client **MUST** send a final SIF_ServiceInput message to the service with the SIF_Error element populated to indicate the nature of the error, and the SIF_MorePackets element set to indicate that no further packets will be sent in for the SIF_ServiceInput.

The SIF_ServiceInput message also contains SIF_Version elements that specify which SIF versions the responding agent should use when preparing the response packets.

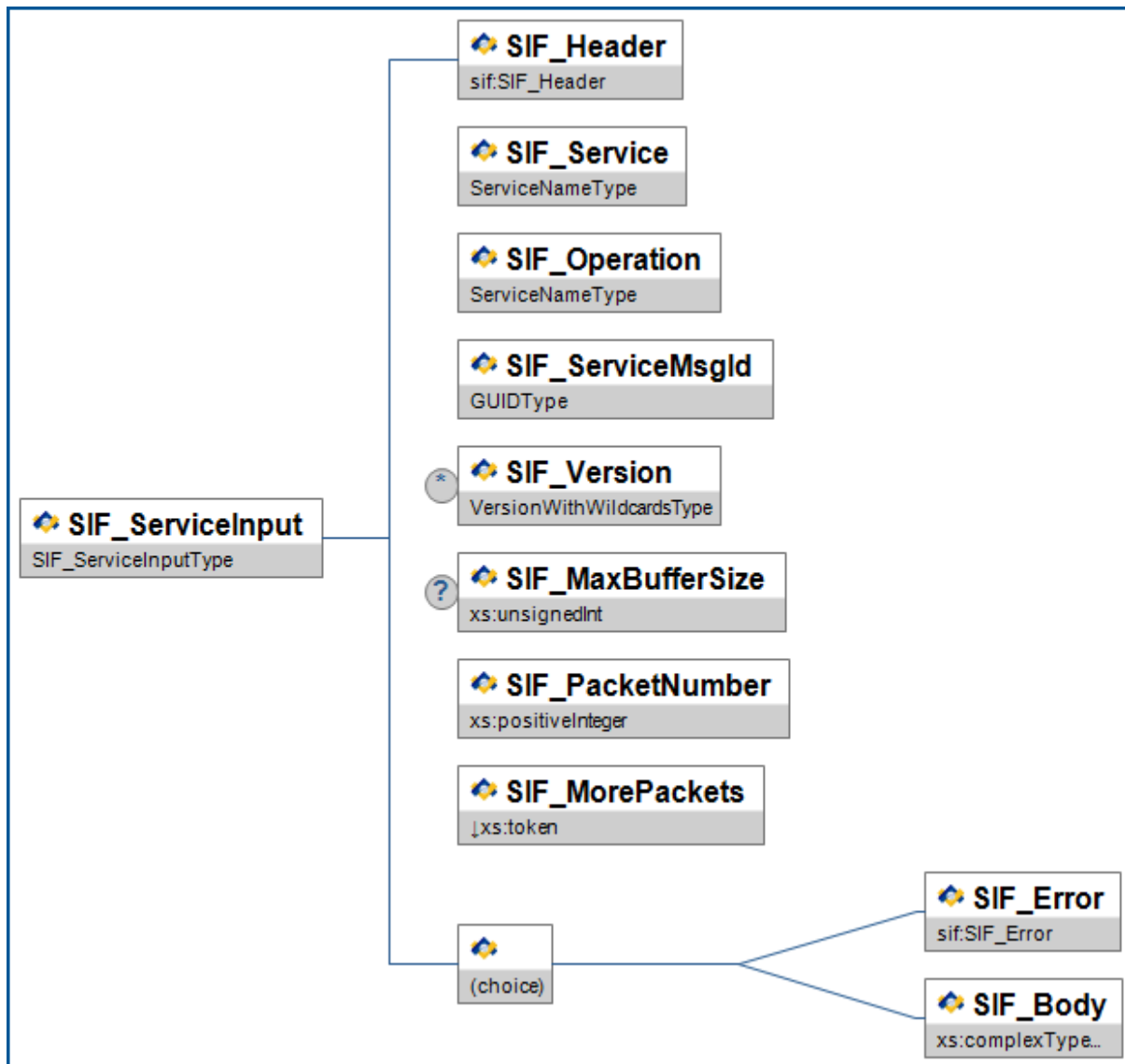


Figure 5.2.21-1: SIF_ServiceInput

Element/@Attribute	Char	Description	Type
SIF_ServiceInput	M	SIF_ServiceInput is used to invoke a method that is exposed by a SIF Zone Service.	
SIF_Header	M	Header information associated with this message.	SIF_Header
SIF_Service	M	The name of the SIF Zone Service that is being invoked.	ServiceNameType
SIF_Operation	M	This is the name of the service operation that is being invoked.	ServiceNameType

Element/@Attribute	Char	Description	Type
SIF_ServiceMsgId	M	The unique Id of this service request invocation. Multiple SIF_ServiceInput packets may be sent with this same SIF_ServiceMsgId	GUIDType
SIF_Version	CR	<p>Specifies which SIF Specification version should be used when returning the response data. If a responder cannot return response data in this format, it should reject the SIF_ServiceInput. It is recommended that clients use a wildcard version for the “minor” portion of the version, such as "2.*".</p> <p>This element is mandatory for the first SIF_ServiceInput packet (SIF_PacketNumber is set to "1"). For subsequent packets, it is not required, and should be ignored, if present, for packets other than the first packet.</p>	VersionWithWildcardsType
SIF_MaxBufferSize	C	<p>Specifies the maximum size of a response packet to be returned to the requester. The responder may return packets smaller than, or equal to, the maximum value. If the maximum size is too small to contain a single whole response object, the responder should send an error back to the requester that the SIF_MaxBufferSize isn't supported. To guarantee delivery of response packets, requesting agents must not specify a SIF_MaxBufferSize greater than its registered SIF_Register/SIF_MaxBufferSize.</p> <p>This element is mandatory for the first SIF_ServiceInput packet (SIF_PacketNumber is set to "1"). For subsequent packets, it is not required, and should be ignored, if present, for packets other than the first packet.</p>	xs:unsignedInt
SIF_PacketNumber	M	<p>This element represents the index of the SIF_ServiceInput message in the sequence of packets that make up a complete response. Its value must be in the range of 1 through n, with n equal to the total number of packets that make up a response.</p> <p>The receiver of a SIF_ServiceInput message, with the help of the SIF_MorePackets and SIF_PacketNumber element in each incoming SIF_ServiceOutput message, will be able to interpret and process each SIF_ServiceInput as part of a complete invocation of a SIF Zone Service Operation.</p>	xs:positiveInteger

Element/@Attribute	Char	Description	Type
SIF_MorePackets	M	<p>This element provides an indication as to whether there are more packets besides this one to make up a complete service request. The value of this element can only be "Yes" or "No".</p> <p>The necessity of this element stems from the requirement on an agent to break service response data into multiple packets to fit into the SIF_MaxBufferSize that has been registered by the service. Agents may also break response data into multiple packets for the benefit of improving performance or for circumventing limitations of the underlying network infrastructure.</p> <p>When this element's value is equal to "No", it is an indication from the sender to the receiver that it has already sent out all the packets that make up a complete SIF_ServiceInput as indicated by the SIF_ServiceMsgId element.</p>	values: Yes No
SIF_Error	C	<p>The agent creates either a SIF_Error or SIF_Body element. The SIF_Error element allows the agent that is invoking the service to report an error condition that occurs while creating the SIF_ServiceInput. Reporting a SIF_Error in SIF_ServiceInput is normally only expected if one or more packets have already been sent so that the receiving agent is aware that an error has occurred that will halt the service invocation message packets. However, a SIF Zone Service definition may define other valid reasons for sending a SIF_Error as the first packet of a SIF_ServiceInput message.</p> <p>If a SIF_Error element is present, the service must not expect to receive further SIF_ServiceInput message.</p>	SIF_Error
SIF_Body	C	<p>SIF_Body contains a single child element that has the same name as the value of the SIF_Operation element. The structure of this element is defined by the XML Schema that is defined for the Service.</p>	<pre><xs:complexType> <xs:sequence> <xs:any processContents="lax" /> </xs:sequence> </xs:complexType></pre>

Table 5.2.21-1: SIF_ServiceInput

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_ServiceInput>
    <SIF_Header>
      <SIF_MsgId>1BCD10580EF250789012AC0554321EA2</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>SISAgent</SIF_SourceId>
      <SIF_DestinationId>NetworkAgent</SIF_DestinationId>
    </SIF_Header>
    <SIF_Service>WeatherService</SIF_Service>
    <SIF_Operation>GetForecast</SIF_Operation>
    <SIF_ServiceMsgId>FE1078BA3261545A319059376B3A4898</SIF_ServiceMsgId>
    <SIF_Version>2.*</SIF_Version>
    <SIF_MaxBufferSize>1048576</SIF_MaxBufferSize>
    <SIF_PacketNumber>1</SIF_PacketNumber>
    <SIF_MorePackets>No</SIF_MorePackets>
    <SIF_Body>
      <GetForecast>
        <PostalCode>55544</PostalCode>
      </GetForecast>
    </SIF_Body>
  </SIF_ServiceInput>
</SIF_Message>

```

Example 5.2.21-1: Example 1 - Simple SIF_ServiceInput

5.2.22 SIF_ServiceOutput

SIF_ServiceOutput is used to respond to a SIF_ServiceInput message. A SIF_ServiceOutput message stream may consist of multiple packets. Each packet consists of a single service element, as defined by the service, contained as a child of the SIF_Body element.

When an agent is creating SIF_ServiceOutput packets, it **MUST** attempt to ensure that each packet is no larger than the SIF_MaxBufferSize specified by the SIF_ServiceInput. If for any packet a single packet does fit within the supplied SIF_MaxBufferSize, the agent **MUST**, in addition to acknowledging receipt of the message to the ZIS, send a SIF_ServiceOutput message to the client with the SIF_Error element populated to indicate the nature of the error, and the SIF_MorePackets element set to indicate that no further packets will be sent in response to the SIF_ServiceInput.

The SIF_ServiceInput message also contains SIF_Version elements that specify which SIF versions the responding agent **SHOULD** use when preparing the response packets. If a responding agent can support a single requested SIF version, it returns a response packet using that version. If more than one version is specified and the responding agent supports more than one of those versions it **SHOULD** respond with the highest version it supports. If the agent cannot support any requested SIF version, it should send a SIF_Error ack back to the ZIS. The ZIS is responsible for constructing a SIF_ServiceOutput error message back to the original agent. The ZIS **MUST** send a SIF_ServiceOutput message to the client with the SIF_Error element populated to indicate the nature of the error, a SIF_PacketNumber of 1 and the SIF_MorePackets element set to indicate that no further packets will be sent in response to the SIF_ServiceInput.

If any other error occurs while creating SIF_ServiceOutput packets for a given request, in addition to acknowledging receipt of the message to the ZIS, the agent **MUST** send a SIF_ServiceOutput message to the client with the SIF_Error element populated to indicate the nature of the error, with SIF_MorePackets set to indicate that no further packets will be sent in response to the SIF_ServiceOutput.

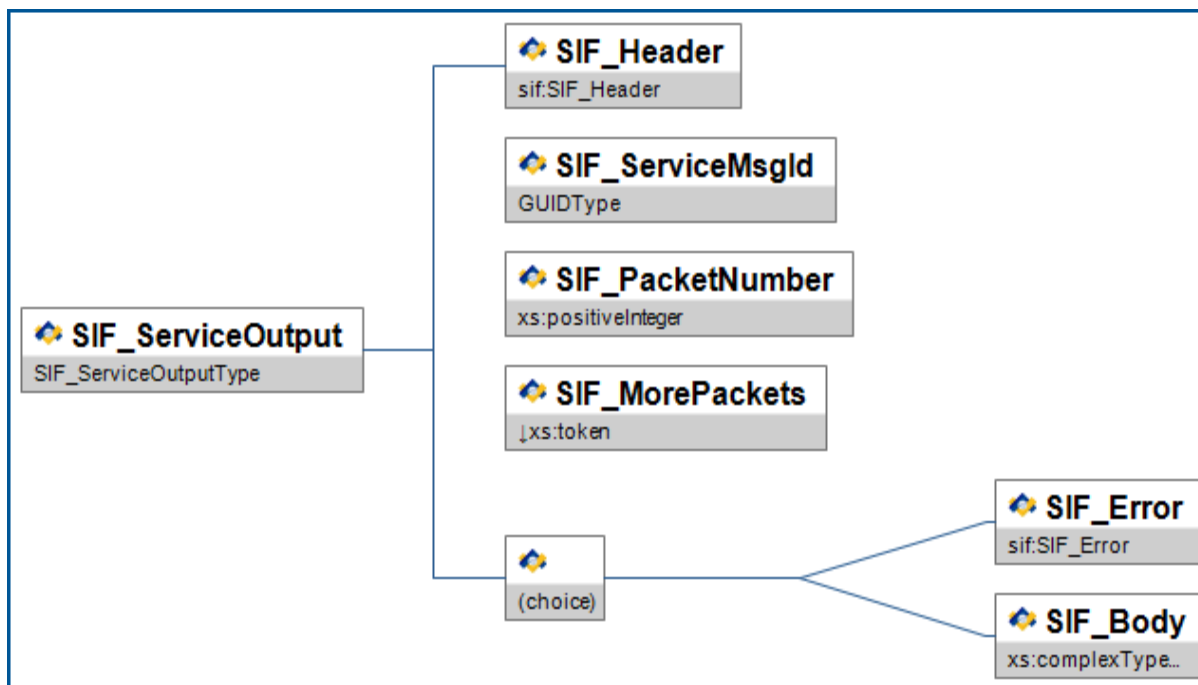


Figure 5.2.22-1: SIF_ServiceOutput

Element/@ Attribute	Char	Description	Type
SIF_ServiceOutput	M	SIF_ServiceOutput is used to respond to a SIF_ServiceInput message.	
SIF_Header	M	Header information associated with this message.	SIF_Header
SIF_ServiceMsgId	M	This element represents the value of the SIF_ServiceMsgId that was sent as part of the original SIF_ServiceInput message stream, and should contain the same value to allow the client and the ZIS to associate each SIF_ServiceOutput packet with the SIF_ServiceInput message stream. This value uniquely identifies the entire set of SIF_ServiceInput and SIF_ServiceOutput messages involved in a single invocation of a SIF Zone Service method.	GUIDType
SIF_PacketNumber	M	<p>This element represents the index of the SIF_ServiceOutput message in the sequence of packets that make up a complete response. Its value must be in the range of 1 through n, with n equal to the total number of packets that make up a response.</p> <p>The receiver of a SIF_ServiceOutput message, with the help of the SIF_MorePackets and SIF_PacketNumber element in each incoming SIF_ServiceOutput message, will be able to interpret and process each SIF_ServiceOutput as part of a complete response to a previous SIF_ServiceInput.</p>	xs:positiveInteger

Element/@Attribute	Char	Description	Type
SIF_MorePackets	M	<p>This element provides an indication as to whether there are more packets besides this one to make up a complete response. The value of this element can only be "Yes" or "No".</p> <p>The necessity of this element stems from the requirement on an agent to break response data to fit into the SIF_MaxBufferSize provided in the SIF_ServiceInput. Agents may also break response data into multiple packets for the benefit of improving performance or for circumventing limitations of the underlying network infrastructure.</p> <p>When this element's value is equal to "No", it is an indication from the sender to the receiver that it has already sent out all the packets that make up a complete response for a SIF_ServiceInput as indicated by the SIF_ServiceMsgId element.</p>	values: Yes No
SIF_Error	C	<p>This element allows the Responder to report an error condition that occurs while processing the SIF_ServiceInput.</p> <p>If a SIF_Error element is present, the requesting agent must not expect to receive further SIF_ServiceOutputs to the SIF_ServiceInput.</p>	SIF_Error
SIF_Body	C	<p>SIF_Body contains a single child element that has a name composed of the value of the SIF_Operation element concatenated with the string "Response". For example a SIF Zone Service Operation with the name "GetWeather" would have in it's response SIF_Body a single child element with the name "GetWeatherResponse". The structure of this element is defined by the XML Schema that is defined for the Service.</p>	<xs:complexType> <xs:sequence> <xs:any processContents="lax" /> </xs:sequence> </xs:complexType>

Table 5.2.22-1: SIF_ServiceOutput

```

<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_ServiceOutput>
    <SIF_Header>
      <SIF_MsgId>1BCD10580EF250789012AC0554321EA2</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>NWS</SIF_SourceId>
      <SIF_DestinationId>ElectronicMarquee</SIF_DestinationId>
    </SIF_Header>
    <SIF_ServiceMsgId>FE1078BA3261545A319059376B3A4898</SIF_ServiceMsgId>
    <SIF_PacketNumber>1</SIF_PacketNumber>
    <SIF_MorePackets>No</SIF_MorePackets>
    <SIF_Body>
      <GetForecastResponse>
        <Sun>Plenty</Sun>
        <Rain>None</Rain>
        <Recommendation>Go To the Beach</Recommendation>
      </GetForecastResponse>
    </SIF_Body>
  </SIF_ServiceOutput>
</SIF_Message>

```

Example 5.2.22-1: SIF_ServiceOutput

5.2.23 SIF_ServiceNotify

SIF_ServiceNotify is a message definition used to deliver service events. Unlike object events, service events may only be issued by the provider of that service.

SIF_ServiceNotify messages **SHOULD** not be sent higher than 64K unless the service knows that all subscribers can accept a higher max buffer size.

Therefore, the size of a SIF_ServiceNotify message **SHOULD** be less than or equal 65,536 bytes, unless one of the following is true:

1. The SIF Zone Service is defined as requiring a larger buffer size for the notification. If that is the case, then the size specified in the SIF Zone Service definition becomes the maximum allowable buffer size.
2. The Service publisher has determined that a higher buffer size is supported by all of the subscribers of the service, either through static configuration by the zone administrator or dynamically at runtime by examination of SIF_ZoneStatus.

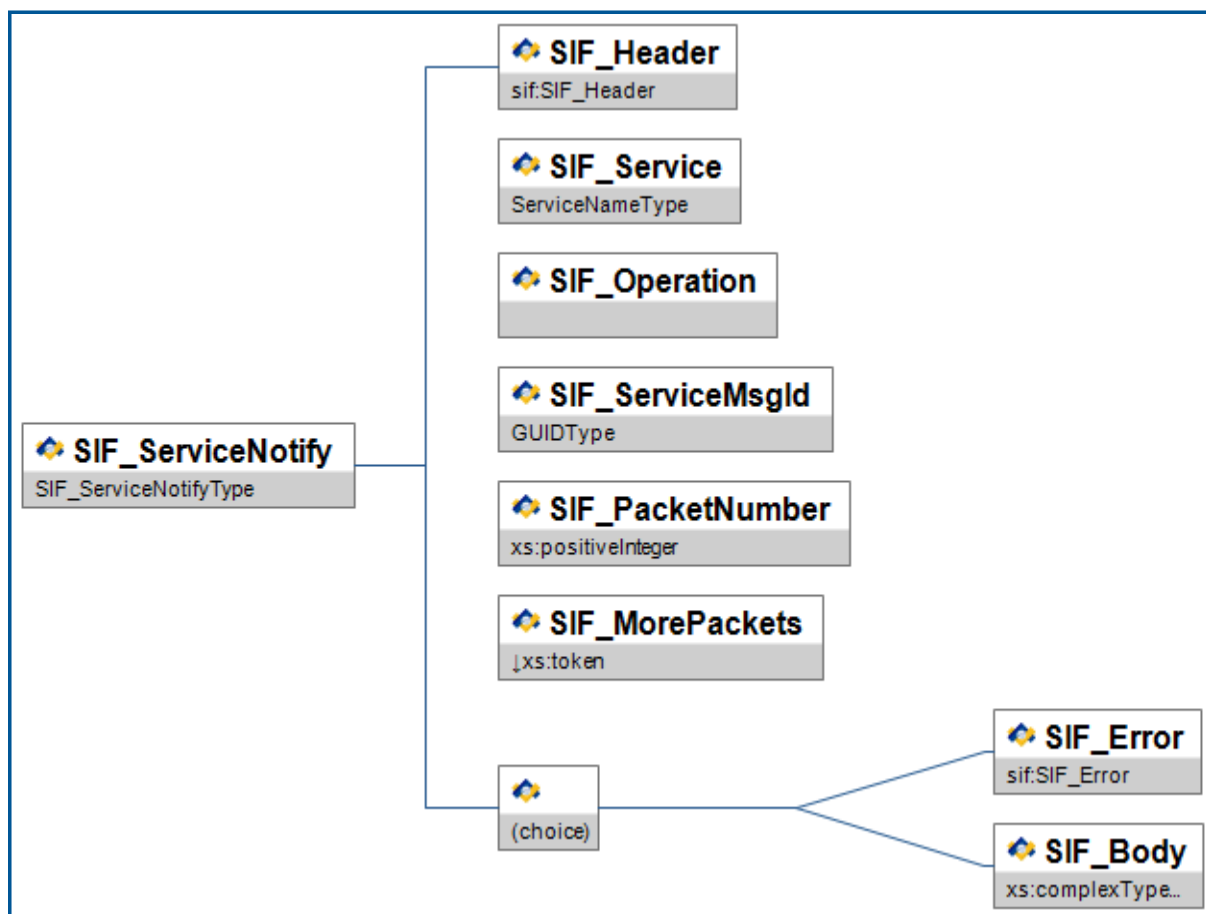


Figure 5.2.23-1: SIF_ServiceNotify

Element/@Attribute	Char	Description	Type
SIF_ServiceNotify	M	SIF_ServiceNotify is used to deliver notifications to service clients that some state associated with the service has changed.	

Element/@ Attribute	Char	Description	Type
SIF_Header	M	Header information associated with this message.	<code>SIF_Header</code>
SIF_Service	M	The name of the SIF Zone Service that initiated the event	<code>ServiceNameType</code>
SIF_Operation	M	The name of the notification message being sent	
SIF_ServiceMsgId	M	A GUID that has been assigned to this series of messages. All SIF_ServiceNotify messages that contain this Id correspond to the same event instance.	<code>GUIDType</code>
SIF_PacketNumber	M	<p>This element represents the index of the SIF_ServiceNotify message in the sequence of packets that make up a complete notification message stream. Its value must be in the range of 1 through n, with n equal to the total number of packets that make up the message stream.</p> <p>The receiver of a SIF_ServiceNotify message, with the help of the SIF_MorePackets and SIF_PacketNumber element in each incoming SIF_ServiceNotify message, will be able to interpret and process each SIF_ServiceNotify as part of a complete message.</p>	<code>xs:positiveInteger</code>
SIF_MorePackets	M	<p>This element provides an indication as to whether there are more packets besides this one to make up a complete notification message stream. The value of this element can only be "Yes" or "No".</p> <p>The necessity of this element stems from the requirement on an agent to break the notification message stream to fit into the SIF_MaxBufferSize specified for the service. Agents may also break the message stream into multiple packets for the benefit of improving performance or for circumventing limitations of the underlying network infrastructure.</p> <p>When this element's value is equal to "No", it is an indication from the sender to the receiver that it has already sent out all of the packets.</p>	values: <code>Yes</code> <code>No</code>

Element/@ Attribute	Char	Description	Type
SIF_Error	C	<p>The agent creates either a SIF_Error or SIF_Body element. The SIF_Error element allows the agent that creating the notification to report an error condition that occurs while creating the SIF_ServiceNotify. Reporting a SIF_Error in SIF_ServiceNotify is normally only expected if one or more packets have already been sent so that the receiving agent is aware that an error has occurred that will halt the notification message packets. However, a SIF Zone Service definition may define other valid reasons for sending a SIF_Error as the first packet of a SIF_ServiceNotify message.</p> <p>If a SIF_Error element is present, the receiver must not expect to receive further SIF_ServiceNotify messages as part of this message stream.</p>	SIF_Error
SIF_Body	C	<p>SIF_Body contains a single child element that has the same name as the value of the SIF_Operation element. The structure of this element is defined by the XML Schema that is defined for the Service.</p>	<pre><xs:complexType> <xs:sequence> <xs:any processContents="lax" /> </xs:sequence> </xs:complexType></pre>

Table 5.2.23-1: SIF_ServiceNotify

```
<SIF_Message Version="2.5" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_ServiceNotify>
    <SIF_Header>
      <SIF_MsgId>1BCD10580EF250789012AC0554321EA2</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>FoodServiceAgent</SIF_SourceId>
    </SIF_Header>
    <SIF_Service>SIS-Service</SIF_Service>
    <SIF_Operation>StudentTransfer</SIF_Operation>
    <SIF_ServiceMsgId>FE1078BA3261545A319059376B3A4898</SIF_ServiceMsgId>
    <SIF_PacketNumber>1</SIF_PacketNumber>
    <SIF_MorePackets>No</SIF_MorePackets>
    <SIF_Body>
      <StudentTransfer>
        <AuditInfo>
          <EnteredBy>sif://StaffPersonal[@RefId='9...12']</EnteredBy>
        </AuditInfo>
        <StudentPersonal RefId="D3E34B359D75101A8C3D00AA001A1652" />
        <PreviousEnrollment>
          <StudentSchoolEnrollment RefId="DFEAD3E34B359D75101D00AA001A1652" />
        </PreviousEnrollment>
        <CurrentEnrollment>
          <StudentSchoolEnrollment RefId="A8C3D3E34B359D75101D00AA001A1652" />
        </CurrentEnrollment>
      </StudentTransfer>
    </SIF_Body>
  </SIF_ServiceNotify>
</SIF_Message>
```

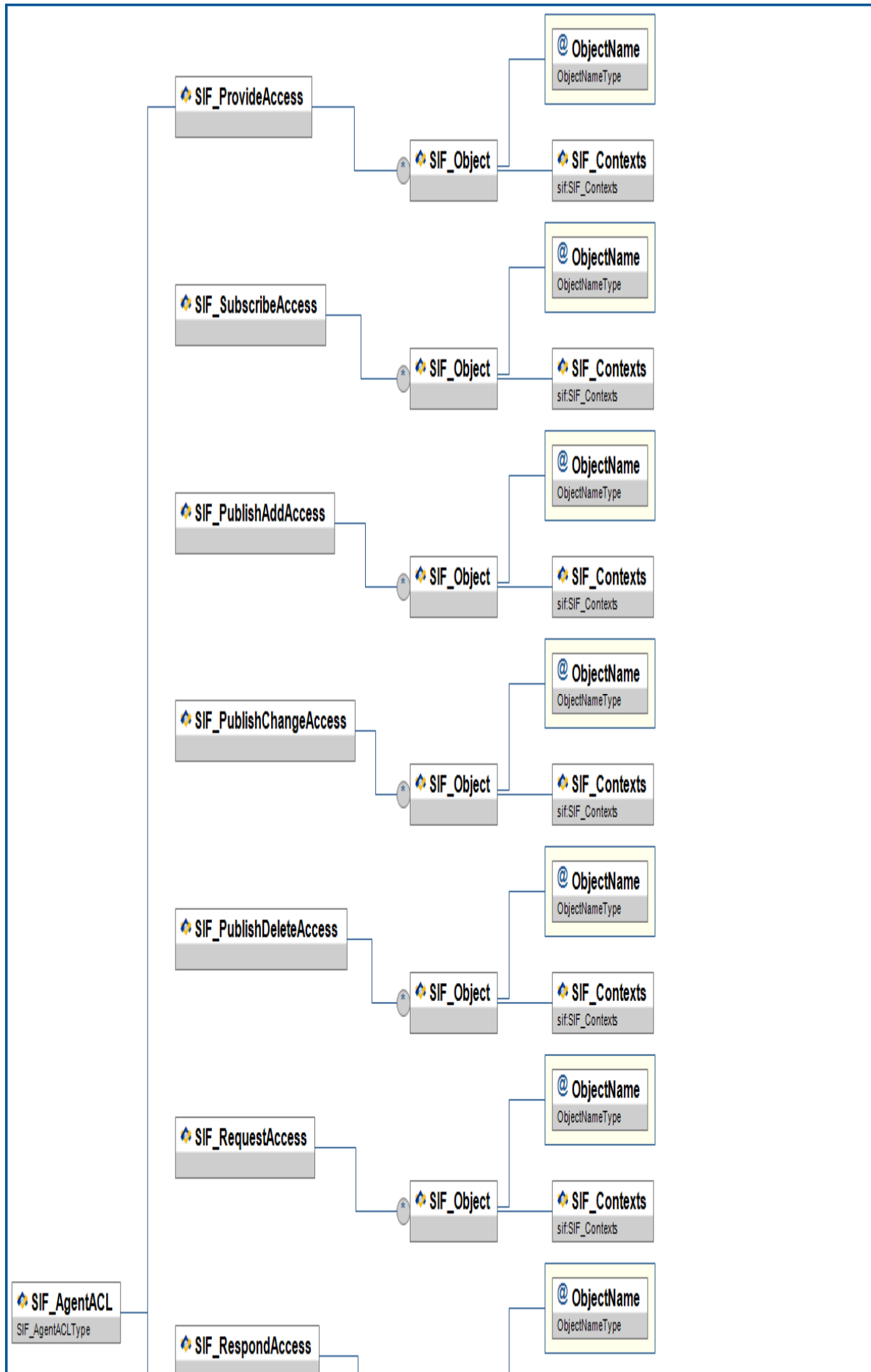
Example 5.2.23-1: SIF_ServiceNotify

5.3 Objects

5.3.1 SIF_AgentACL

This object provides an Agent its access control list (ACL) settings in the Zone. It does not communicate which objects the Agent is currently registered as providing, subscribing, publishing, requesting, or responding; it simply lists the ACL rights granted to the Agent in the Zone. When objects are absent from any of the access lists, the Agent does not have the necessary rights to perform the given action on the object. While an Agent may asynchronously request this object from the ZIS via `SIF_Request`, it is typically returned synchronously in response to the `SIF_SystemControl` message `SIF_GetAgentACL`; it is also returned synchronously in response to `SIF_Register`.

Typically only `Change` events are reported.



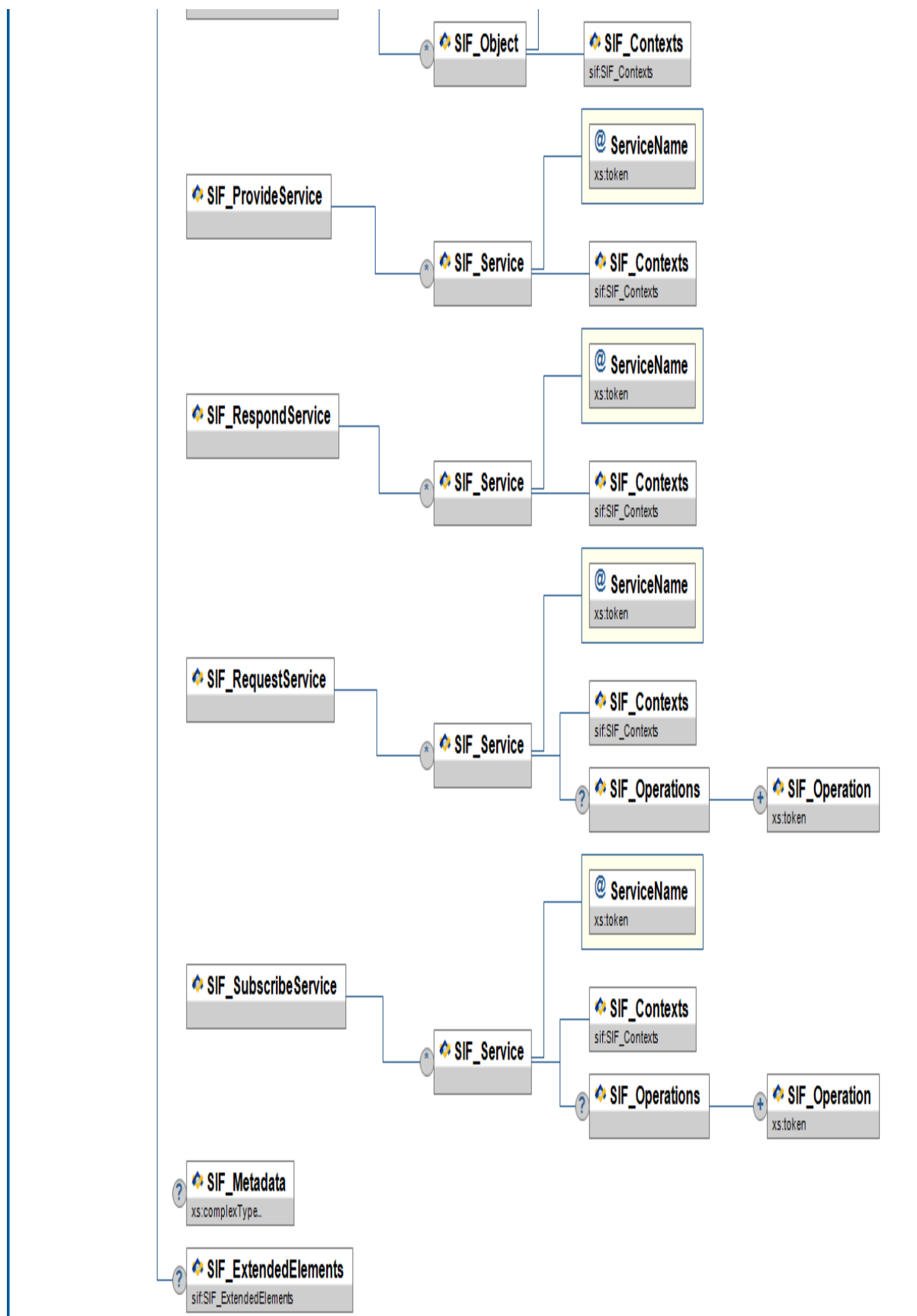


Figure 5.3.1-1: SIF_AgentACL

	Element/@Attribute	Char	Description	Type
	SIF_AgentACL		This object provides an Agent its access control list (ACL) settings in the Zone.	
	SIF_ProvideAccess	M	Access control list by object for SIF_Provide and corresponding section in SIF_Provision.	List
	SIF_ProvideAccess/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_ProvideAccess/SIF_Object/ SIF_Contexts	M	List of contexts in which rights for the given action/object apply.	SIF_Contexts
	SIF_SubscribeAccess	M	Access control list by object for SIF_Subscribe and corresponding section in SIF_Provision.	List
	SIF_SubscribeAccess/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_SubscribeAccess/SIF_Object/ SIF_Contexts	M	List of contexts in which rights for the given action/object apply.	SIF_Contexts
	SIF_PublishAddAccess	M	Access control list by object for corresponding section in SIF_Provision, whether the Agent has the right to publish Add events.	List
	SIF_PublishAddAccess/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType

	Element/@Attribute	Char	Description	Type
	SIF_PublishAddAccess/SIF_Object/ SIF_Contexts	M	List of contexts in which rights for the given action/object apply.	SIF_Contexts
	SIF_PublishChangeAccess	M	Access control list by object for corresponding section in SIF_Provision, whether the Agent has the right to publish Change events.	List
	SIF_PublishChangeAccess/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_PublishChangeAccess/SIF_Object/ SIF_Contexts	M	List of contexts in which rights for the given action/object apply.	SIF_Contexts
	SIF_PublishDeleteAccess	M	Access control list by object for corresponding section in SIF_Provision, whether the Agent has the right to publish Delete events.	List
	SIF_PublishDeleteAccess/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_PublishDeleteAccess/SIF_Object/ SIF_Contexts	M	List of contexts in which rights for the given action/object apply.	SIF_Contexts
	SIF_RequestAccess	M	Access control list by object for SIF_Request and corresponding section in SIF_Provision.	List
	SIF_RequestAccess/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType

	Element/@Attribute	Char	Description	Type
	SIF_RequestAccess/SIF_Object/ SIF_Contexts	M	List of contexts in which rights for the given action/object apply.	SIF_Contexts
	SIF_RespondAccess	M	Access control list by object for corresponding section in SIF_Provision, whether the Agent has the right to respond to requests for an object regardless of being the Provider of that object.	List
	SIF_RespondAccess/SIF_Object	OR		
@	ObjectName	M	The name of each object.	ObjectNameType
	SIF_RespondAccess/SIF_Object/ SIF_Contexts	M	List of contexts in which rights for the given action/object apply.	SIF_Contexts
	SIF_ProvideService	M	Indicates that the recipient agent has permission to provide one or more services to the SIF Zone	List
	SIF_ProvideService/SIF_Service	OR		
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	xs:token
	SIF_ProvideService/SIF_Service/ SIF_Contexts	M	List of contexts in which rights for the given action/service apply.	SIF_Contexts
	SIF_RespondService	M	Indicates that the recipient agent has permission to respond to directed requests for one or more services in the SIF Zone	List
	SIF_RespondService/SIF_Service	OR		

	Element/@Attribute	Char	Description	Type
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	<code>xs:token</code>
	SIF_RespondService/SIF_Service/ SIF_Contexts	M	List of contexts in which rights for the given action/service apply.	<code>SIF_Contexts</code>
	SIF_RequestService	M	Indicates that the recipient agent has permission to make service calls to a SIF Zone Service	<code>List</code>
	SIF_RequestService/SIF_Service	OR		
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	<code>xs:token</code>
	SIF_RequestService/SIF_Service/ SIF_Contexts	M	List of contexts in which rights for the given action/service apply.	<code>SIF_Contexts</code>
	SIF_RequestService/SIF_Service/ SIF_Operations	O	If SIF_Operations is not present, then the agent has permission to invoke all operations in the specified service.	<code>List</code>
	SIF_RequestService/SIF_Service/ SIF_Operations/SIF_Operation	MR	A specific operation that the agent has permission to invoke or subscribe to	<code>xs:token</code>
	SIF_SubscribeService	M	Indicates that the recipient agent has permission to subscribe to notification messages that are emitted from a SIF Service	<code>List</code>
	SIF_SubscribeService/SIF_Service	OR		

	Element/@Attribute	Char	Description	Type
@	ServiceName	M	The name of the SIF Zone Service as defined by a SIF Zone Service specification	xs:token
	SIF_SubscribeService/SIF_Service/SIF_Contexts	M	List of contexts in which rights for the given action/service apply.	SIF_Contexts
	SIF_SubscribeService/SIF_Service/SIF_Operations	O	If SIF_Operations is not present, then the agent has permission to subscribe to all notifications in the specified service .	List
	SIF_SubscribeService/SIF_Service/SIF_Operations/SIF_Operation	MR	A specific operation that the agent has permission to invoke or subscribe to	xs:token
	SIF_Metadata	O		<pre><xs:complexType> <xs:sequence> <xs:any minOccurs="0" maxOccurs="unbounded" /> </xs:sequence> </xs:complexType></pre>
	SIF_ExtendedElements	O		SIF_ExtendedElements

Table 5.3.1-1: SIF_AgentACL


```

<SIF_AgentACL>
  <SIF_ProvideAccess>
    <SIF_Object ObjectName="StudentPersonal">
      <SIF_Contexts>
        <SIF_Context>SIF_Default</SIF_Context>
      </SIF_Contexts>
    </SIF_Object>
  </SIF_ProvideAccess>
  <SIF_SubscribeAccess>
    <SIF_Object ObjectName="Authentication">
      <SIF_Contexts>
        <SIF_Context>SIF_Default</SIF_Context>
      </SIF_Contexts>
    </SIF_Object>
  </SIF_SubscribeAccess>
  <SIF_PublishAddAccess>
    <SIF_Object ObjectName="StudentPersonal">
      <SIF_Contexts>
        <SIF_Context>SIF_Default</SIF_Context>
      </SIF_Contexts>
    </SIF_Object>
  </SIF_PublishAddAccess>
  <SIF_PublishChangeAccess>
    <SIF_Object ObjectName="StudentPersonal">
      <SIF_Contexts>
        <SIF_Context>SIF_Default</SIF_Context>
      </SIF_Contexts>
    </SIF_Object>
  </SIF_PublishChangeAccess>
  <SIF_PublishDeleteAccess>
    <SIF_Object ObjectName="StudentPersonal">
      <SIF_Contexts>
        <SIF_Context>SIF_Default</SIF_Context>
      </SIF_Contexts>
    </SIF_Object>
  </SIF_PublishDeleteAccess>
  <SIF_RequestAccess>
    <SIF_Object ObjectName="Authentication">
      <SIF_Contexts>
        <SIF_Context>SIF_Default</SIF_Context>
      </SIF_Contexts>
    </SIF_Object>
  </SIF_RequestAccess>
  <SIF_RespondAccess>
    <SIF_Object ObjectName="StudentPersonal">
      <SIF_Contexts>
        <SIF_Context>SIF_Default</SIF_Context>
      </SIF_Contexts>
    </SIF_Object>
  </SIF_RespondAccess>
</SIF_AgentACL>

```

Example 5.3.1-1: SIF_AgentACL

5.3.2 SIF_LogEntry

This object captures an occurrence within a SIF node (ZIS or agent)—error, warning or information—for storage in an optionally provided zone log. SIF_LogEntry Adds are reported and are used to post new log entries to the provider of the log. Of course, subscribing agents may also filter incoming Adds as part of their own logging mechanism. Any Change or Delete SIF_Events should be ignored at the agent level, but should be routed by the ZIS (though this should not be necessary). Use of the log is optional and voluntary, except where noted as mandatory in this specification. Nodes may post as much or as little log data as required with the expectation that if there is a provider of SIF_LogEntry that the logged entries be available for a provider-defined amount of time subject to provider-defined restrictions on the quantity of data logged by any given node.

SIF_Events are reported for this object.

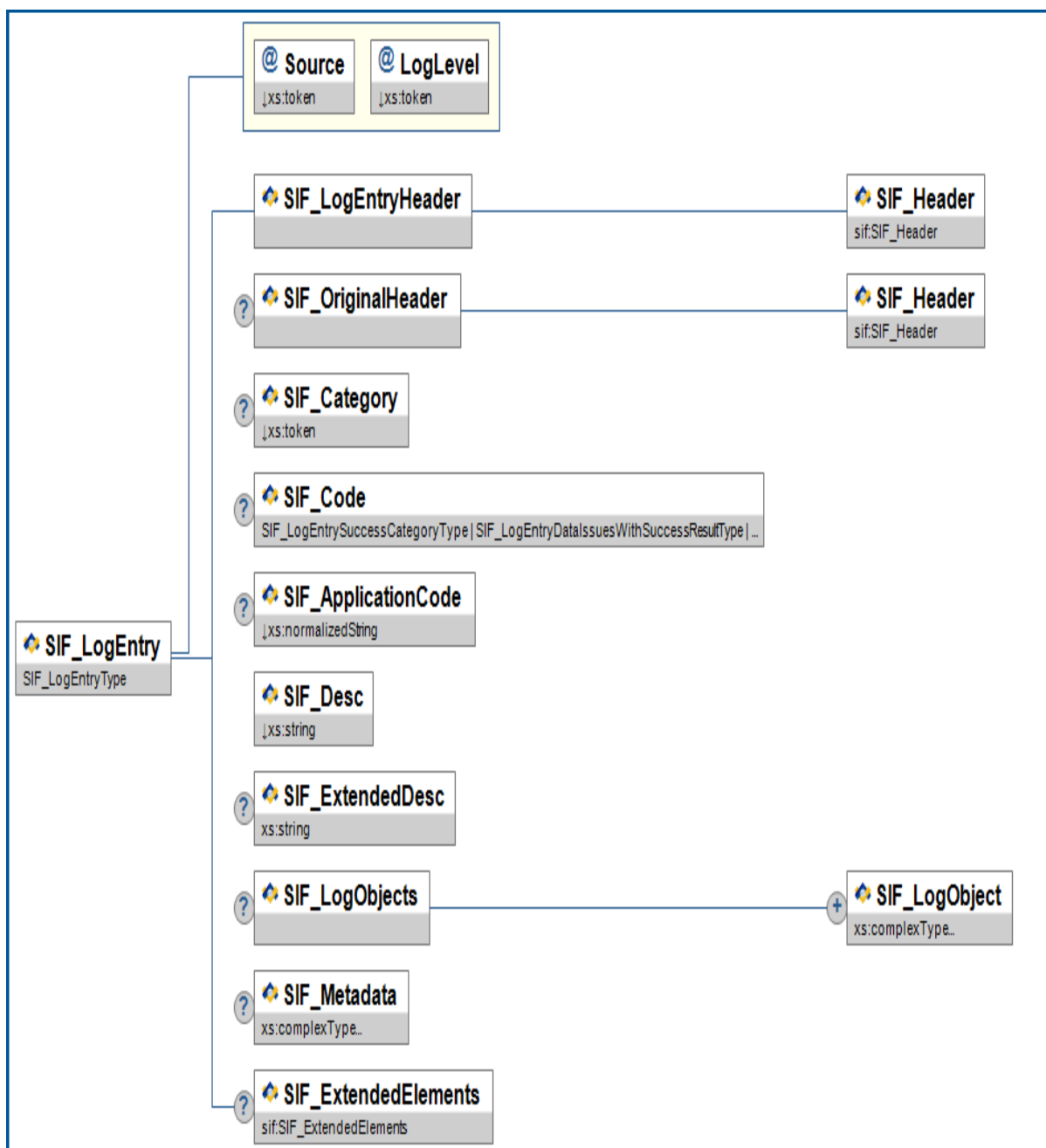


Figure 5.3.2-1: SIF_LogEntry

	Element/@Attribute	Char	Description	Type
	SIF_LogEntry		<p>This object captures an occurrence within a SIF node (ZIS or agent)—error, warning or information—for storage in an optionally provided zone log.</p> <p>SIF_LogEntry Adds are reported and are used to post new log entries to the provider of the log. Of course, subscribing agents may also filter incoming Adds as part of their own logging mechanism. Any Change or Delete SIF_Events should be ignored at the agent level, but should be routed by the ZIS (though this should not be necessary). Use of the log is optional and voluntary, except where noted as mandatory in this specification. Nodes may post as much or as little log data as required with the expectation that if there is a provider of SIF_LogEntry that the logged entries be available for a provider-defined amount of time subject to provider-defined restrictions on the quantity of data logged by any given node.</p>	
@	Source	M	<p>The SIF node that logged this entry.</p>	<p>values:</p> <p>Agent ZIS</p>
@	LogLevel	M	<p>The level of the log entry herein described.</p>	<p>values:</p> <p>Info Warning Error</p>

Element/@Attribute	Char	Description	Type
SIF_LogEntryHeader	M	This is a copy of the SIF_Event/SIF_Header in the message that added this SIF_LogEntry to the zone. This copy facilitates querying log entries with regard to source, time, optionally destination, etc.	
SIF_LogEntryHeader/SIF_Header	M		SIF_Header
SIF_OriginalHeader	O	If this log entry references a previous SIF_Message, this element contains a copy of the referenced message's SIF_Header.	
SIF_OriginalHeader/SIF_Header	M		SIF_Header
SIF_Category	C	<p>A SIF_LogEntry category. May be omitted for informational-type postings, where typically a textual description will suffice.</p> <p>Note that categories may be combined with the Source attribute of SIF_LogEntry to differentiate agent error conditions from ZIS error conditions.</p>	values: <ol style="list-style-type: none"> 1 Success 2 Data Issues with Success Result 3 Data Issues with Failure Result 4 Error Conditions
SIF_Code	O	A SIF_LogEntry code with regard to SIF_Category above. May be omitted for informational-type postings, where typically a textual description will suffice. If a SIF_Code is included, SIF_Category must be included as well.	union of: <ul style="list-style-type: none"> SIF_LogEntrySuccessCategoryType SIF_LogEntryDataIssuesWithSuccessResultType SIF_LogEntryDataIssuesWithFailureResultType SIF_LogEntryAgentErrorConditionType SIF_LogEntryZISErrorConditionType

Element/@Attribute	Char	Description	Type
SIF_ApplicationCode	O	An error code specific to the application posting the entry. Can be used by vendors to query log entries for errors specific to their applications. If a SIF_ApplicationCode is included, SIF_Category must be included as well; i.e., application-specific error codes should fall within one of the defined log entry categories.	<code>xs:normalizedString</code> <div> <div>xs:maxLength</div> <div>64</div> </div>
SIF_Desc	M	A textual description of the error.	<code>xs:string</code> <div> <div>xs:maxLength</div> <div>1024</div> </div>
SIF_ExtendedDesc	O	Any extended error description.	<code>xs:string</code>
SIF_LogObjects	O		List
SIF_LogObjects/SIF_LogObject	MR	Any SIF data objects to which this log entry may apply.	<pre> <xs:complexType> <xs:sequence> <xs:any processContents="skip" /> </xs:sequence> <xs:attribute name="ObjectName" use="required" type="xs:NCName" /> </xs:complexType> </pre>
@ ObjectName	M	The name of the SIF object referenced (e.g. StudentPersonal).	ObjectNameType
SIF_Metadata	O		<pre> <xs:complexType> <xs:sequence> <xs:any minOccurs="0" maxOccurs="unbounded" /> </xs:sequence> </xs:complexType> </pre>
SIF_ExtendedElements	O		SIF_ExtendedElements

Table 5.3.2-1: SIF_LogEntry

```
<SIF_LogEntry Source="Agent" LogLevel="Error">
  <SIF_LogEntryHeader>
    <SIF_Header>
      <SIF_MsgId>83252CE5C5F14FD88607F645224E4CAA</SIF_MsgId>
      <SIF_Timestamp>2006-08-19T10:36:00-05:00</SIF_Timestamp>
      <SIF_SourceId>RamseySISAgent</SIF_SourceId>
    </SIF_Header>
  </SIF_LogEntryHeader>
  <SIF_Category>4</SIF_Category>
  <SIF_Code>1</SIF_Code>
  <SIF_Desc>Agent has run out of memory and will shut down</SIF_Desc>
  <SIF_ExtendedDesc>OutOfMemoryException: ...</SIF_ExtendedDesc>
</SIF_LogEntry>
```

Example 5.3.2-1: SIF_LogEntry when an agent encounters a system failure

```

<SIF_LogEntry Source="Agent" LogLevel="Error">
  <SIF_LogEntryHeader>
    <SIF_Header>
      <SIF_MsgId>BA86894B795A4EB7A45093AD1CDBA54C</SIF_MsgId>
      <SIF_Timestamp>2006-08-19T10:39:00-05:00</SIF_Timestamp>
      <SIF_SourceId>RamseySISAgent</SIF_SourceId>
    </SIF_Header>
  </SIF_LogEntryHeader>
  <SIF_OriginalHeader>
    <SIF_Header>
      <SIF_MsgId>74234DCB460A4BCB8937B07467EA73CC</SIF_MsgId>
      <SIF_Timestamp>2006-08-19T10:29:00-05:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLibraryAgent</SIF_SourceId>
    </SIF_Header>
  </SIF_OriginalHeader>
  <SIF_Category>3</SIF_Category>
  <SIF_Code>2</SIF_Code>
  <SIF_ApplicationCode>-33</SIF_ApplicationCode>
  <SIF_Desc>Could not delete student John Smith due to business rule</SIF_Desc>
  <SIF_LogObjects>
    <SIF_LogObject ObjectName="StudentPersonal">
      <StudentPersonal RefId="76D3A70232FE40D7A5D43A7A317EAEF9">
        <AlertMessages>
          <AlertMessage Type="Legal">This is the Legal Alert for Joe Student</AlertMessage>
        </AlertMessages>
        <LocalId>P00001</LocalId>
        <StateProvinceId>WB0025</StateProvinceId>
        <ElectronicIdList>
          <ElectronicId Type="Barcode">206654</ElectronicId>
        </ElectronicIdList>
        <Name Type="04">
          <LastName>Student</LastName>
          <FirstName>Joe</FirstName>
          <MiddleName />
          <PreferredName>Joe</PreferredName>
        </Name>
        <Demographics>
          <Gender>M</Gender>
        </Demographics>
        <AddressList>
          <Address Type="0123">
            <Street>
              <Line1>6799 33rd Ave.</Line1>
              <StreetNumber>6799</StreetNumber>
              <StreetName>33rd</StreetName>
              <StreetType>Ave.</StreetType>
            </Street>
            <City>Chicago</City>
            <StateProvince>IL</StateProvince>
            <Country>US</Country>
            <PostalCode>60660</PostalCode>
          </Address>
        </AddressList>
        <PhoneNumberList>
          <PhoneNumber Type="0096">
            <Number>(312) 555-1234</Number>
          </PhoneNumber>
        </PhoneNumberList>
        <EmailList>
          <Email Type="Primary">Joe.Student@anyschool.com</Email>
        </EmailList>
        <OnTimeGraduationYear>2007</OnTimeGraduationYear>
      </StudentPersonal>
    </SIF_LogObject>
  </SIF_LogObjects>
</SIF_LogEntry>

```

Example 5.3.2-2: SIF_LogEntry when an agent fails to delete a student

```

<SIF_LogEntry Source="Agent" LogLevel="Info">
  <SIF_LogEntryHeader>
    <SIF_Header>
      <SIF_MsgId>64B0CC6CFB314A328E520A102229CBC8</SIF_MsgId>
      <SIF_Timestamp>2006-08-19T10:46:00-05:00</SIF_Timestamp>
      <SIF_SourceId>RamseySISAgent</SIF_SourceId>
    </SIF_Header>
  </SIF_LogEntryHeader>
  <SIF_Desc>Agent starting synchronization</SIF_Desc>
</SIF_LogEntry>

```

Example 5.3.2-3: SIF_LogEntry when an agent starts synchronizing data

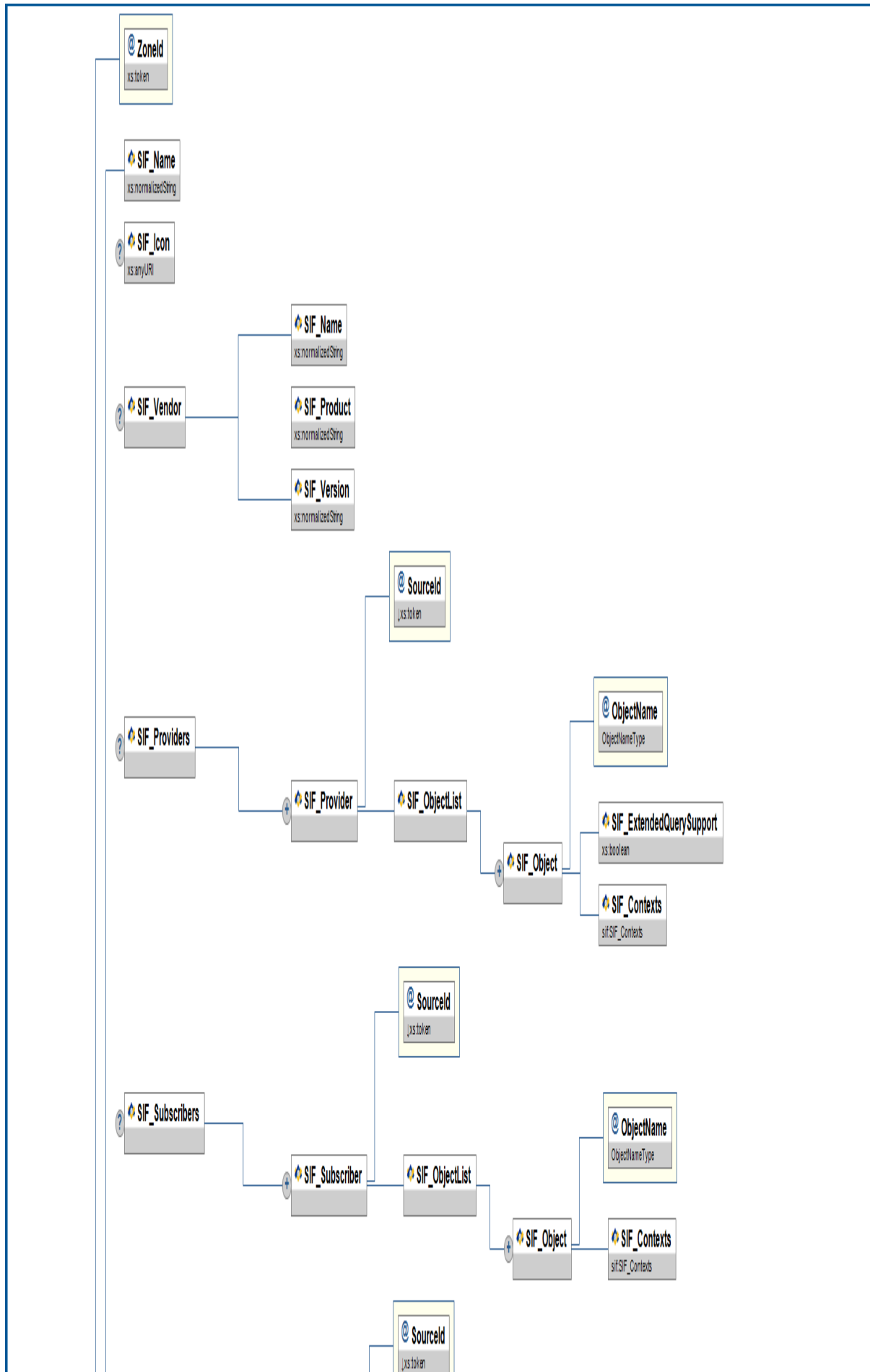
```
<SIF_LogEntry Source="ZIS" LogLevel="Error">
  <SIF_LogEntryHeader>
    <SIF_Header>
      <SIF_MsgId>BC1D982CEC5F49D998169930FE5B271C</SIF_MsgId>
      <SIF_Timestamp>2006-08-19T10:49:00-05:00</SIF_Timestamp>
      <SIF_SourceId>RamseyZIS</SIF_SourceId>
    </SIF_Header>
  </SIF_LogEntryHeader>
  <SIF_OriginalHeader>
    <SIF_Header>
      <SIF_MsgId>74234DCB460A4BCB8937B07467EA73CC</SIF_MsgId>
      <SIF_Timestamp>2006-08-19T10:29:00-05:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLibraryAgent</SIF_SourceId>
    </SIF_Header>
  </SIF_OriginalHeader>
  <SIF_Category>4</SIF_Category>
  <SIF_Code>2</SIF_Code>
  <SIF_Desc>Could not deliver StudentPicture Add to RamseyLibraryAgent (127,546 bytes) due to maximum buffer size
</SIF_LogEntry>
```

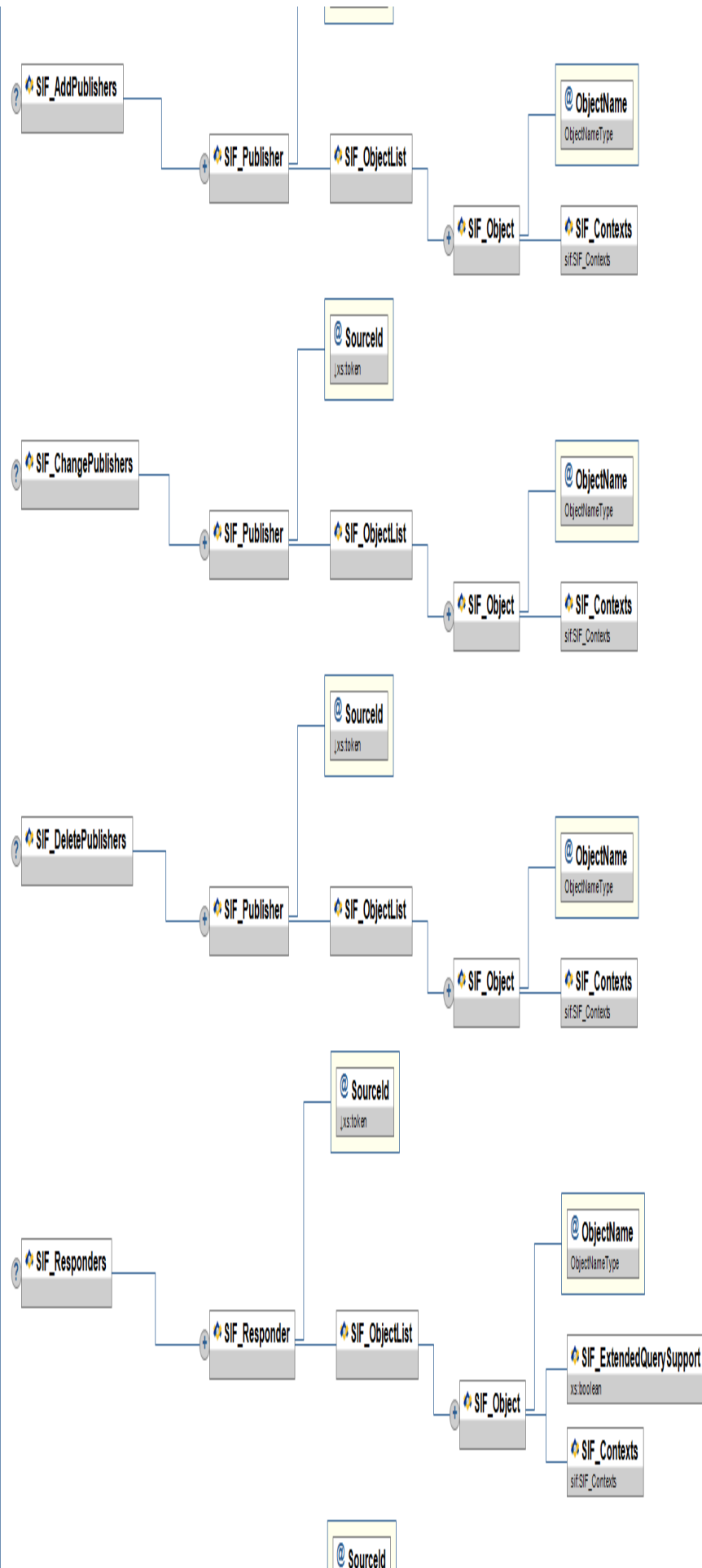
Example 5.3.2-4: SIF_LogEntry when a ZIS fails to deliver a message due to buffer size limitations

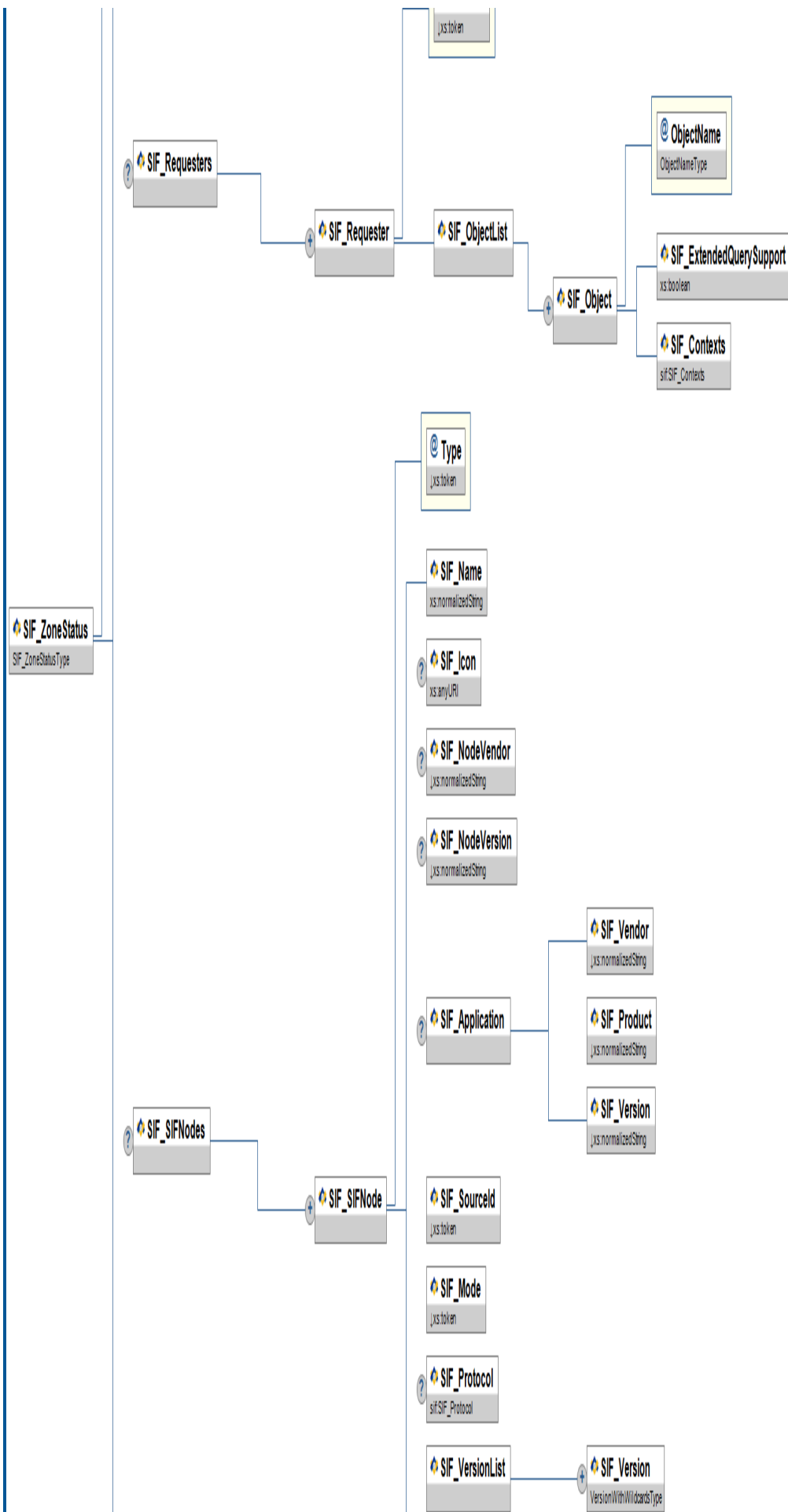
5.3.3 SIF_ZoneStatus

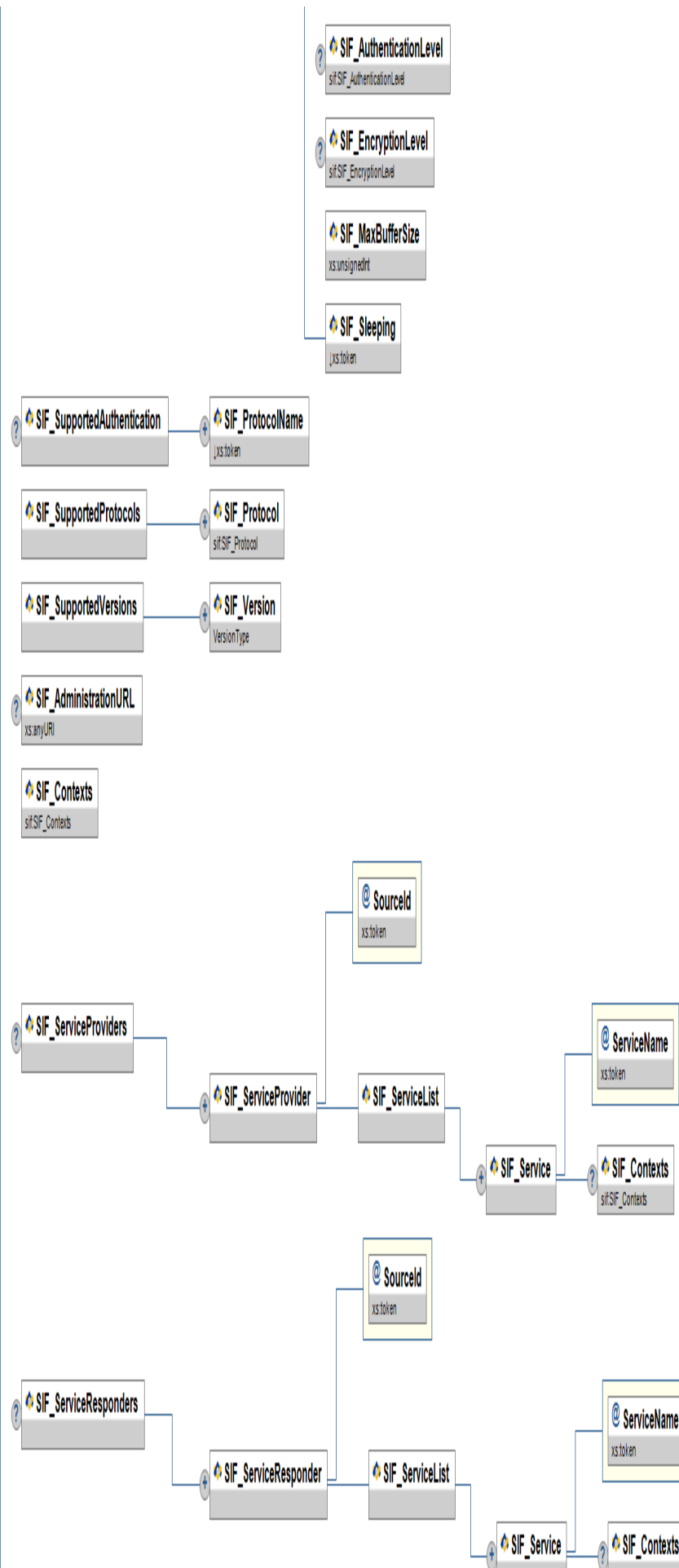
The `SIF_ZoneStatus` object is an object that is implicitly provided by all Zone Integration Servers to provide information about the ZIS. Zone Integration Servers **MUST** provide this object.

Change events are supported on `SIF_ZoneStatus`.









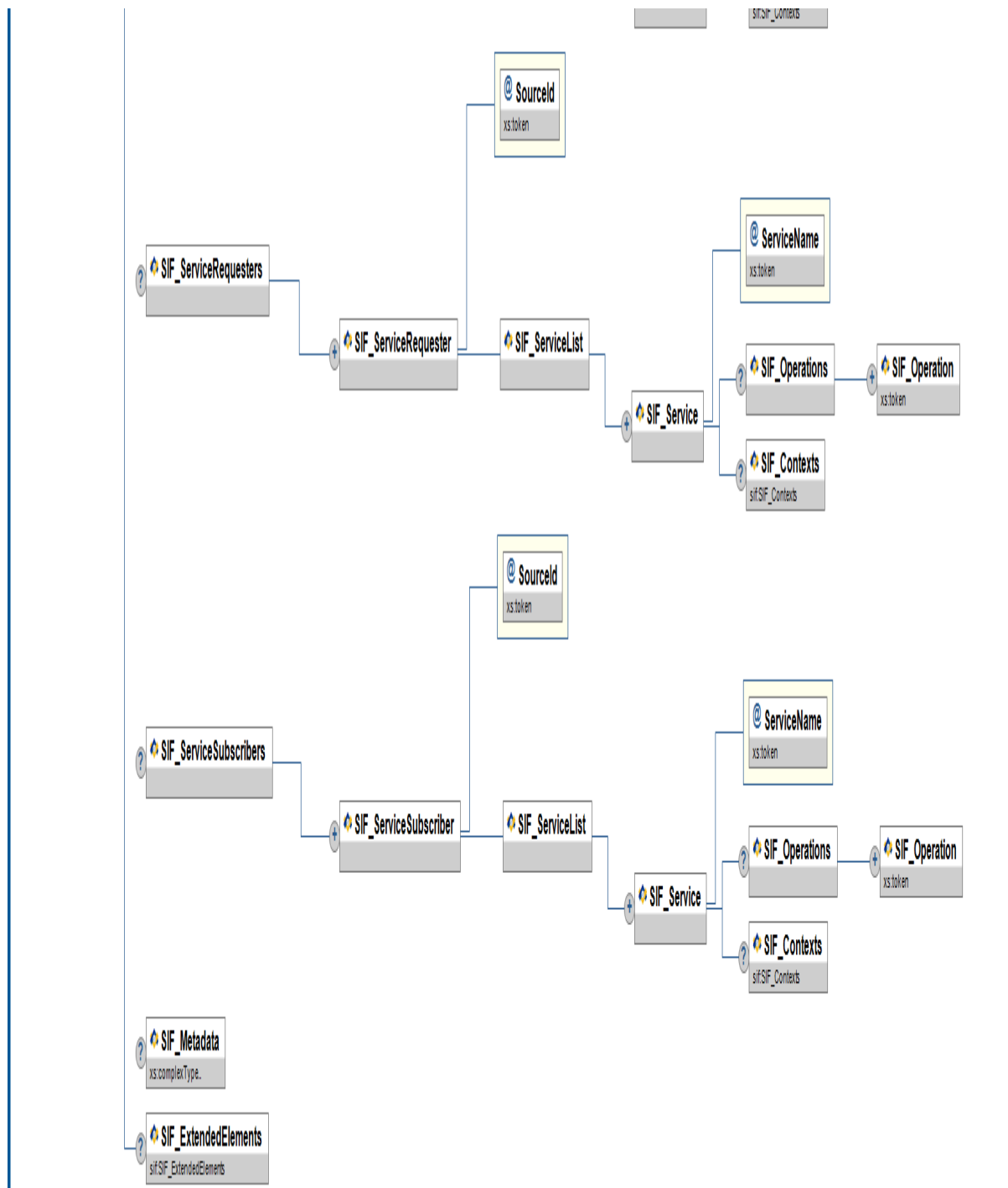



Figure 5.3.3-1: SIF_ZoneStatus

Element/@ Attribute	Char	Description	Type
SIF_ZoneStatus		The <code>SIF_ZoneStatus</code> object is an object that is implicitly provided by all Zone Integration Servers to provide information about the ZIS. Zone Integration Servers MUST provide this object.	

	Element/@ Attribute	Char	Description	Type
@ 	ZoneId	M	The identifier for this Zone. It is the same as the <code>SIF_SourceId</code> that the ZIS would place in any <code>SIF_Header</code> that it creates.	<code>xs:token</code>
	SIF_Name	M	The descriptive name for the zone.	<code>xs:normalizedString</code>
	SIF_Icon	O	HTTP URL referencing an icon for graphical representation of the ZIS/Zone. Should range from 16x16 pixels to 128x128 pixels and be of an <code>image</code> MIME type commonly supported by Web browsers (e.g. PNG, JPEG, GIF). Agents may optionally follow the more restrictive guidelines at [FAVICON] .	<code>xs:anyURI</code>
	SIF_Vendor	O	Contains information about the vendor that wrote this ZIS.	
	SIF_Vendor/SIF_Name	M	The name of the company that wrote the ZIS.	<code>xs:normalizedString</code>
	SIF_Vendor/SIF_Product	M	The product name assigned by the vendor to identify this ZIS.	<code>xs:normalizedString</code>
	SIF_Vendor/SIF_Version	M	The version of the vendor's product—not necessarily the SIF version.	<code>xs:normalizedString</code>

	Element/@ Attribute	Char	Description	Type
	SIF_Providers	C	Encompasses all the providers registered with this ZIS. This element is mandatory if there are providers registered with the ZIS.	List
	SIF_Providers/SIF_Provider	MR		
@	SourceId	M	The identifier of the SIF node that is providing objects. This is the agent or ZIS identifier that would appear in the <code>SIF_SourceId</code> field of any <code>SIF_Header</code> created by the SIF node.	<code>xs:token</code> <div> <code>xs:maxLength</code> 64 </div>
	SIF_Providers/SIF_Provider/SIF_ObjectList	M		List
	SIF_Providers/SIF_Provider/SIF_ObjectList/SIF_Object	MR		
@	ObjectName	M	The name of the object being provided by this SIF node.	ObjectNameType
	SIF_Providers/SIF_Provider/SIF_ObjectList/SIF_Object/SIF_ExtendedQuerySupport	M		<code>xs:boolean</code>
	SIF_Providers/SIF_Provider/SIF_ObjectList/SIF_Object/SIF_Contexts	M		SIF_Contexts
	SIF_Subscribers	C	Encompasses all the subscribers registered with this ZIS. This element is mandatory if there are subscribers registered with the ZIS.	List
	SIF_Subscribers/SIF_Subscriber	MR		

	Element/@ Attribute	Char	Description	Type
@	SourceId	M	The identifier of the SIF node that is subscribing to the object events. This is the agent or ZIS identifier that would appear in the <code>SIF_SourceId</code> field of any <code>SIF_Header</code> created by the SIF node.	<code>xs:token</code> <div> <code>xs:maxLength</code> 64 </div>
	SIF_Subscribers/SIF_Subscriber/SIF_ObjectList	M		List
	SIF_Subscribers/SIF_Subscriber/SIF_ObjectList/SIF_Object	MR		
@	ObjectName	M	The name of the object being subscribed to by this SIF node.	<code>ObjectNameType</code>
	SIF_Subscribers/SIF_Subscriber/SIF_ObjectList/SIF_Object/SIF_Contexts	M		<code>SIF_Contexts</code>
	SIF_AddPublishers	C	Encompasses all the <code>Add SIF_Event</code> publishers registered with this zone.	List
	SIF_AddPublishers/SIF_Publisher	MR		
@	SourceId	M	The identifier of the SIF node that can publish the <code>SIF_Event</code> . This is the agent identifier that would appear in the <code>SIF_SourceId</code> field of any <code>SIF_Header</code> created by the agent.	<code>xs:token</code> <div> <code>xs:maxLength</code> 64 </div>
	SIF_AddPublishers/SIF_Publisher/SIF_ObjectList	M		List
	SIF_AddPublishers/SIF_Publisher/SIF_ObjectList/SIF_Object	MR		
@	ObjectName	M	The name of the object being published by this agent.	<code>ObjectNameType</code>
	SIF_AddPublishers/SIF_Publisher/SIF_ObjectList/SIF_Object/SIF_Contexts	M		<code>SIF_Contexts</code>

	Element/@ Attribute	Char	Description	Type
	SIF_ChangePublishers	C	Encompasses all the Change SIF_Event publishers registered with this zone.	List
	SIF_ChangePublishers/SIF_Publisher	MR		
@	SourceId	M	The identifier of the SIF node that can publish the SIF_Event. This is the agent identifier that would appear in the SIF_SourceId field of any SIF_Header created by the agent.	xs:token <div>xs:maxLength64</div>
	SIF_ChangePublishers/SIF_Publisher/ SIF_ObjectList	M		List
	SIF_ChangePublishers/SIF_Publisher/ SIF_ObjectList/SIF_Object	MR		
@	ObjectName	M	The name of the object being published by this agent.	ObjectNameType
	SIF_ChangePublishers/SIF_Publisher/ SIF_ObjectList/SIF_Object/ SIF_Contexts	M		SIF_Contexts
	SIF_DeletePublishers	C	Encompasses all the Delete SIF_Event publishers registered with this zone.	List
	SIF_DeletePublishers/SIF_Publisher	MR		
@	SourceId	M	The identifier of the SIF node that can publish the SIF_Event. This is the agent identifier that would appear in the SIF_SourceId field of any SIF_Header created by the agent.	xs:token <div>xs:maxLength64</div>
	SIF_DeletePublishers/SIF_Publisher/ SIF_ObjectList	M		List
	SIF_DeletePublishers/SIF_Publisher/ SIF_ObjectList/SIF_Object	MR		

	Element/@ Attribute	Char	Description	Type
@	ObjectName	M	The name of the object being published by this agent.	ObjectNameType
	SIF_DeletePublishers/SIF_Publisher/ SIF_ObjectList/SIF_Object/ SIF_Contexts	M		SIF_Contexts
	SIF_Responders	C	Encompasses all the responders registered with this zone.	List
	SIF_Responders/SIF_Responder	MR		
@	SourceId	M	The identifier of the SIF node that can respond. This is the agent identifier that would appear in the SIF_SourceId field of any SIF_Header created by the agent.	xs:token <div>xs:maxLength 64</div>
	SIF_Responders/SIF_Responder/ SIF_ObjectList	M		List
	SIF_Responders/SIF_Responder/ SIF_ObjectList/SIF_Object	MR		
@	ObjectName	M	The name of the object for which the agent can respond to requests.	ObjectNameType
	SIF_Responders/SIF_Responder/ SIF_ObjectList/SIF_Object/ SIF_ExtendedQuerySupport	M		xs:boolean
	SIF_Responders/SIF_Responder/ SIF_ObjectList/SIF_Object/ SIF_Contexts	M		SIF_Contexts
	SIF_Requesters	C	Encompasses all the requesters registered with this zone.	List
	SIF_Requesters/SIF_Requester	MR		

	Element/@ Attribute	Char	Description	Type
@	SourceId	M	The identifier of the SIF node that can request an object. This is the agent identifier that would appear in the <code>SIF_SourceId</code> field of any <code>SIF_Header</code> created by the agent.	<code>xs:token</code> <div> <code>xs:maxLength</code> 64 </div>
	SIF_Requesters/SIF_Requester/SIF_ObjectList	M		List
	SIF_Requesters/SIF_Requester/SIF_ObjectList/SIF_Object	MR		
@	ObjectName	M	The name of the object being requested by this agent.	ObjectNameType
	SIF_Requesters/SIF_Requester/SIF_ObjectList/SIF_Object/SIF_ExtendedQuerySupport	M		<code>xs:boolean</code>
	SIF_Requesters/SIF_Requester/SIF_ObjectList/SIF_Object/SIF_Contexts	M		SIF_Contexts
	SIF_SIFNodes	C	Encompasses all of the nodes registered with the ZIS. This element is mandatory if there are SIF nodes registered.	List
	SIF_SIFNodes/SIF_SIFNode	MR		
@	Type	M	The type of the node registered with the ZIS. Note that <code>ZIS</code> is forward-looking and not used currently; all information about this Zone/ZIS is contained outside <code>SIF_SIFNodes</code> .	values: Agent ZIS
	SIF_SIFNodes/SIF_SIFNode/SIF_Name	M	The descriptive name of the SIF node (i.e. Ramsey Food Services).	<code>xs:normalizedString</code>

Element/@ Attribute	Char	Description	Type
SIF_SIFNodes/SIF_SIFNode/ SIF_Icon	O	HTTP URL referencing an icon for graphical representation of the application/agent. Should range from 16x16 pixels to 128x128 pixels and be of an <code>image</code> MIME type commonly supported by Web browsers (e.g. PNG, JPEG, GIF). Agents may optionally follow the more restrictive guidelines at [FAVICON] .	<code>xs:anyURI</code>
SIF_SIFNodes/SIF_SIFNode/ SIF_NodeVendor	O	The vendor of the SIF agent.	<code>xs:normalizedString</code> <div><code>xs:maxLength</code> 256</div>
SIF_SIFNodes/SIF_SIFNode/ SIF_NodeVersion	O	The agent version number. The format of this field is undefined, but it should match the format used in the agent's conformance statement, if the agent is SIF Certified. Examples 2.0.1.11	<code>xs:normalizedString</code> <div><code>xs:maxLength</code> 32</div>
SIF_SIFNodes/SIF_SIFNode/ SIF_Application	O	Contains information about the vendor of the product that the agent represents.	
SIF_SIFNodes/SIF_SIFNode/ SIF_Application/SIF_Vendor	M	The name of the company of the product that this agent supports.	<code>xs:normalizedString</code> <div><code>xs:maxLength</code> 256</div>
SIF_SIFNodes/SIF_SIFNode/ SIF_Application/SIF_Product	M	The name of the product that this agent supports.	<code>xs:normalizedString</code> <div><code>xs:maxLength</code> 256</div>
SIF_SIFNodes/SIF_SIFNode/ SIF_Application/SIF_Version	M	The version of the product. This field is informative only.	<code>xs:normalizedString</code> <div><code>xs:maxLength</code> 32</div>

Element/@ Attribute	Char	Description	Type
SIF_SIFNodes/SIF_SIFNode/ SIF_SourceId	M	The agent or ZIS identifier. This is the same value that the SIF node would place in any SIF_Header that it would create.	xs:token <div>xs:maxLength64</div>
SIF_SIFNodes/SIF_SIFNode/ SIF_Mode	M	Specifies the communication mode (Pull or Push) as chosen by the message sender.	values: Push Pull
SIF_SIFNodes/SIF_SIFNode/ SIF_Protocol	O	Describes the currently active protocol that the SIF node is using to communicate with the ZIS.	SIF_Protocol
SIF_SIFNodes/SIF_SIFNode/ SIF_VersionList	M		List
SIF_SIFNodes/SIF_SIFNode/ SIF_VersionList/SIF_Version	MR	This is the version or versions of the SIF Implementation Specification that define(s) the messages the SIF node can receive. For agents, this information was communicated when the SIF node registered with the ZIS.	VersionWithWildcardsType
SIF_SIFNodes/SIF_SIFNode/ SIF_AuthenticationLevel	O	This is the level of authentication that the SIF node supports when it wants to communicate via a secure channel.	SIF_AuthenticationLevel
SIF_SIFNodes/SIF_SIFNode/ SIF_EncryptionLevel	O	This is the level of encryption that the SIF node supports when it wants to communicate via a secure channel.	SIF_EncryptionLevel

Element/@ Attribute	Char	Description	Type
SIF_SIFNodes/SIF_SIFNode/ SIF_MaxBufferSize	M	Specifies that the ZIS should never send packets larger than this value. Query responses from other providers are controlled by the SIF_MaxBufferSize attribute in the SIF_Request message.	xs:unsignedInt
SIF_SIFNodes/SIF_SIFNode/ SIF_Sleeping	M	This element shows whether the SIF node is ready to process messages.	values: No The SIF node is ready to process messages Yes The SIF node is sleeping and cannot process messages
SIF_SupportedAuthentication	C	Enumerates the various authentication protocols that the ZIS supports. If the ZIS supports an authentication protocol this element is mandatory.	List
SIF_SupportedAuthentication/ SIF_ProtocolName	MR	Describes a particular authentication protocol supported.	values: X.509
SIF_SupportedProtocols	M	Enumerates the various communication transport protocols that are supported by the ZIS.	List
SIF_SupportedProtocols/SIF_Protocol	MR		SIF_Protocol
SIF_SupportedVersions	M	Enumerates the versions of the SIF Implementation Specification that this ZIS can use when communicating with the agent.	List
SIF_SupportedVersions/SIF_Version	MR	Lists a specific SIF Implementation Specification version.	VersionType

	Element/@ Attribute	Char	Description	Type
	SIF_AdministrationURL	O	Should a ZIS vendor provide an administration interface for the zone via a URL, the ZIS can make the URL available in SIF_ZoneStatus. Agent administrators can use the URL to access zone administration features, should they have permission to do so.	xs:anyURI
	SIF_Contexts	M		SIF_Contexts
	SIF_ServiceProviders	O		List
	SIF_ServiceProviders/SIF_ServiceProvider	MR	A list of nodes within a SIF Zone that provide one or more SIF Zone Services. The provider of a SIF Zone Service can be a SIF Agent or the Zone Integration Server (ZIS) itself.	
@	SourceId	M	The identifier of the SIF node that is providing SIF Services. This is the agent or ZIS identifier that would appear in the SIF_SourceId field of any SIF_Header created by the SIF node.	xs:token
	SIF_ServiceProviders/SIF_ServiceProvider/SIF_ServiceList	M	The list of services provided by this node	List
	SIF_ServiceProviders/SIF_ServiceProvider/SIF_ServiceList/SIF_Service	MR		
@	ServiceName		The name of the SIF Zone Service as defined by a SIF Zone Service specification	xs:token

	Element/@ Attribute	Char	Description	Type
	SIF_ServiceProviders/SIF_ServiceProvider/ SIF_ServiceList/SIF_Service/ SIF_Contexts	O	Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to SIF_Default.	SIF_Contexts
	SIF_ServiceResponders	O	A list of nodes within a SIF Zone that will respond to SIF_ServiceInput messages for one or more SIF Zone Services. The responder can be a SIF Agent or the Zone Integration Server (ZIS) itself.	List
	SIF_ServiceResponders/SIF_ServiceResponder	MR		
@	SourceId	M	The identifier of the SIF node that is providing SIF Services. This is the agent or ZIS identifier that would appear in the SIF_SourceId field of any SIF_Header created by the SIF node.	xs:token
	SIF_ServiceResponders/SIF_ServiceResponder/ SIF_ServiceList	M	The list of services that will be responded to by this node.	List
	SIF_ServiceResponders/SIF_ServiceResponder/ SIF_ServiceList/SIF_Service	MR		
@	ServiceName		The name of the SIF Zone Service as defined by a SIF Zone Service specification	xs:token
	SIF_ServiceResponders/SIF_ServiceResponder/ SIF_ServiceList/SIF_Service/ SIF_Contexts	O	Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to SIF_Default.	SIF_Contexts

	Element/@ Attribute	Char	Description	Type
	SIF_ServiceRequesters	O	A list of nodes within a SIF Zone that will respond to SIF_ServiceInput messages for one or more SIF Zone Services. The responder can be a SIF Agent or the Zone Integration Server (ZIS) itself.	List
	SIF_ServiceRequesters/SIF_ServiceRequester	MR		
@	SourceId	M	The identifier of the SIF node that will respond to SIF_ServiceInput messages. This is the agent or ZIS identifier that would appear in the SIF_SourceId field of any SIF_Header created by the SIF node.	xs:token
	SIF_ServiceRequesters/SIF_ServiceRequester/SIF_ServiceList	M	The list of services that will be invoked by this node	List
	SIF_ServiceRequesters/SIF_ServiceRequester/SIF_ServiceList/SIF_Service	MR		
@	ServiceName		The name of the SIF Zone Service as defined by a SIF Zone Service specification	xs:token

	Element/@ Attribute	Char	Description	Type
	SIF_ServiceRequesters/SIF_ServiceRequester/ SIF_ServiceList/SIF_Service/ SIF_Operations	O	The list of operations an agent may invoke on a SIF Zone Service. This information may or may not be known by the ZIS as it is optionally provided by an agent during SIF_Provision. The list of operations an agent may invoke on a SIF Zone Service. This information may or may not be known by the ZIS as it is optionally provided by an agent during SIF_Provision.	List
	SIF_ServiceRequesters/SIF_ServiceRequester/ SIF_ServiceList/SIF_Service/ SIF_Operations/SIF_Operation	MR	A specific operation with a SIF Zone Service that the agent will invoke.	xs:token
	SIF_ServiceRequesters/SIF_ServiceRequester/ SIF_ServiceList/SIF_Service/ SIF_Contexts	O		SIF_Contexts
	SIF_ServiceSubscribers	O		List
	SIF_ServiceSubscribers/SIF_ServiceSubscriber	MR		
@	SourceId	M	The identifier of the SIF node that is providing SIF Services. This is the agent or ZIS identifier that would appear in the SIF_SourceId field of any SIF_Header created by the SIF node.	xs:token
	SIF_ServiceSubscribers/SIF_ServiceSubscriber/ SIF_ServiceList	M	The list of services that are subscribed to by this node.	List
	SIF_ServiceSubscribers/SIF_ServiceSubscriber/ SIF_ServiceList/SIF_Service	MR		

	Element/@ Attribute	Char	Description	Type
@	ServiceName		The name of the SIF Zone Service as defined by a SIF Zone Service specification	<code>xs:token</code>
	SIF_ServiceSubscribers/SIF_ServiceSubscriber/ SIF_ServiceList/SIF_Service/ SIF_Operations	O	If SIF_Operations is not present, then the agent is subscribed to all events emitted by the service	List
	SIF_ServiceSubscribers/SIF_ServiceSubscriber/ SIF_ServiceList/SIF_Service/ SIF_Operations/SIF_Operation	MR	A specific notification message that the agent is subscribed to	<code>xs:token</code>
	SIF_ServiceSubscribers/SIF_ServiceSubscriber/ SIF_ServiceList/SIF_Service/ SIF_Contexts	O		<code>SIF_Contexts</code>
	SIF_Metadata	O		<pre><xs:complexType> <xs:sequence> <xs:any minOccurs="0" maxOccurs="unbounded" /> </xs:sequence> </xs:complexType></pre>
	SIF_ExtendedElements	O		<code>SIF_ExtendedElements</code>

Table 5.3.3-1: SIF_ZoneStatus

```

<SIF_ZoneStatus ZoneId="RamseyZIS">
  <SIF_Name>Ramsey Elementary</SIF_Name>
  <SIF_Vendor>
    <SIF_Name>ZoneMaster, Inc.</SIF_Name>
    <SIF_Product>ZonePlus Zone Integration Server</SIF_Product>
    <SIF_Version>3.01</SIF_Version>
  </SIF_Vendor>
  <SIF_Providers>
    <SIF_Provider SourceId="RamseyFOOD">
      <SIF_ObjectList>
        <SIF_Object ObjectName="StudentMeal">
          <SIF_ExtendedQuerySupport>false</SIF_ExtendedQuerySupport>
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
      </SIF_ObjectList>
    </SIF_Provider>
    <SIF_Provider SourceId="RamseyLIB">
      <SIF_ObjectList>
        <SIF_Object ObjectName="LibraryPatronStatus">
          <SIF_ExtendedQuerySupport>false</SIF_ExtendedQuerySupport>
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
      </SIF_ObjectList>
    </SIF_Provider>
    <SIF_Provider SourceId="RamseySIS">
      <SIF_ObjectList>
        <SIF_Object ObjectName="StudentPersonal">
          <SIF_ExtendedQuerySupport>false</SIF_ExtendedQuerySupport>
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
        <SIF_Object ObjectName="StudentSchoolEnrollment">
          <SIF_ExtendedQuerySupport>false</SIF_ExtendedQuerySupport>
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
      </SIF_ObjectList>
    </SIF_Provider>
  </SIF_Providers>
  <SIF_Subscribers>
    <SIF_Subscriber SourceId="RamseyFOOD">
      <SIF_ObjectList>
        <SIF_Object ObjectName="StudentPersonal">
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
        <SIF_Object ObjectName="StudentSchoolEnrollment">
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
      </SIF_ObjectList>
    </SIF_Subscriber>
    <SIF_Subscriber SourceId="RamseyLIB">
      <SIF_ObjectList>
        <SIF_Object ObjectName="StudentPersonal">
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
        <SIF_Object ObjectName="StudentSchoolEnrollment">
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
      </SIF_ObjectList>
    </SIF_Subscriber>
    <SIF_Subscriber SourceId="RamseySIS">
      <SIF_ObjectList>
        <SIF_Object ObjectName="StudentContact">
          <SIF_Contexts>
            <SIF_Context>SIF_Default</SIF_Context>
          </SIF_Contexts>
        </SIF_Object>
      </SIF_ObjectList>
    </SIF_Subscriber>
  </SIF_Subscribers>

```

```

</SIF_Subscribers>
<SIF_SIFNodes>
  <SIF_SIFNode Type="Agent">
    <SIF_Name>Ramsey Food Services</SIF_Name>
    <SIF_SourceId>RamseyFOOD</SIF_SourceId>
    <SIF_Mode>Push</SIF_Mode>
    <SIF_Protocol Type="HTTPS" Secure="Yes">
      <SIF_URL>https://RamseyNT:8010/FoodService</SIF_URL>
    </SIF_Protocol>
    <SIF_VersionList>
      <SIF_Version>2.3</SIF_Version>
    </SIF_VersionList>
    <SIF_MaxBufferSize>16384</SIF_MaxBufferSize>
    <SIF_Sleeping>No</SIF_Sleeping>
  </SIF_SIFNode>
  <SIF_SIFNode Type="Agent">
    <SIF_Name>Ramsey Media Resource Center</SIF_Name>
    <SIF_SourceId>RamseyLIB</SIF_SourceId>
    <SIF_Mode>Pull</SIF_Mode>
    <SIF_Protocol Type="HTTPS" Secure="Yes">
      <SIF_URL>https://RamseyNT:8020/Library</SIF_URL>
    </SIF_Protocol>
    <SIF_VersionList>
      <SIF_Version>2.3</SIF_Version>
    </SIF_VersionList>
    <SIF_MaxBufferSize>16384</SIF_MaxBufferSize>
    <SIF_Sleeping>No</SIF_Sleeping>
  </SIF_SIFNode>
  <SIF_SIFNode Type="Agent">
    <SIF_Name>Ramsey Administration</SIF_Name>
    <SIF_SourceId>RamseySIS</SIF_SourceId>
    <SIF_Mode>Push</SIF_Mode>
    <SIF_Protocol Type="HTTPS" Secure="Yes">
      <SIF_URL>https://RamseyNT:8030/StudentAdmin</SIF_URL>
    </SIF_Protocol>
    <SIF_VersionList>
      <SIF_Version>2.3</SIF_Version>
    </SIF_VersionList>
    <SIF_MaxBufferSize>16384</SIF_MaxBufferSize>
    <SIF_Sleeping>No</SIF_Sleeping>
  </SIF_SIFNode>
</SIF_SIFNodes>
<SIF_SupportedAuthentication>
  <SIF_ProtocolName>X.509</SIF_ProtocolName>
</SIF_SupportedAuthentication>
<SIF_SupportedProtocols>
  <SIF_Protocol Type="HTTPS" Secure="Yes">
    <SIF_URL>https://RamseyNT:8000/ZIS</SIF_URL>
  </SIF_Protocol>
  <SIF_Protocol Type="HTTP" Secure="No">
    <SIF_URL>http://RamseyNT:8000/ZIS</SIF_URL>
  </SIF_Protocol>
</SIF_SupportedProtocols>
<SIF_SupportedVersions>
  <SIF_Version>2.3</SIF_Version>
</SIF_SupportedVersions>
<SIF_AdministrationURL>http://RamseyNT:8000/Administer</SIF_AdministrationURL>
<SIF_Contexts>
  <SIF_Context>SIF_Default</SIF_Context>
</SIF_Contexts>
</SIF_ZoneStatus>

```

Example 5.3.3-1: SIF_ZoneStatus

Appendix A: Common Types

Common and supporting types referenced in this specification are included here as a reference.

A.1 AbstractContentType

AbstractContentType used as an element rather than an object, omitting **RefId**, **SIF_Metadata** and **SIF_ExtendedElements**.

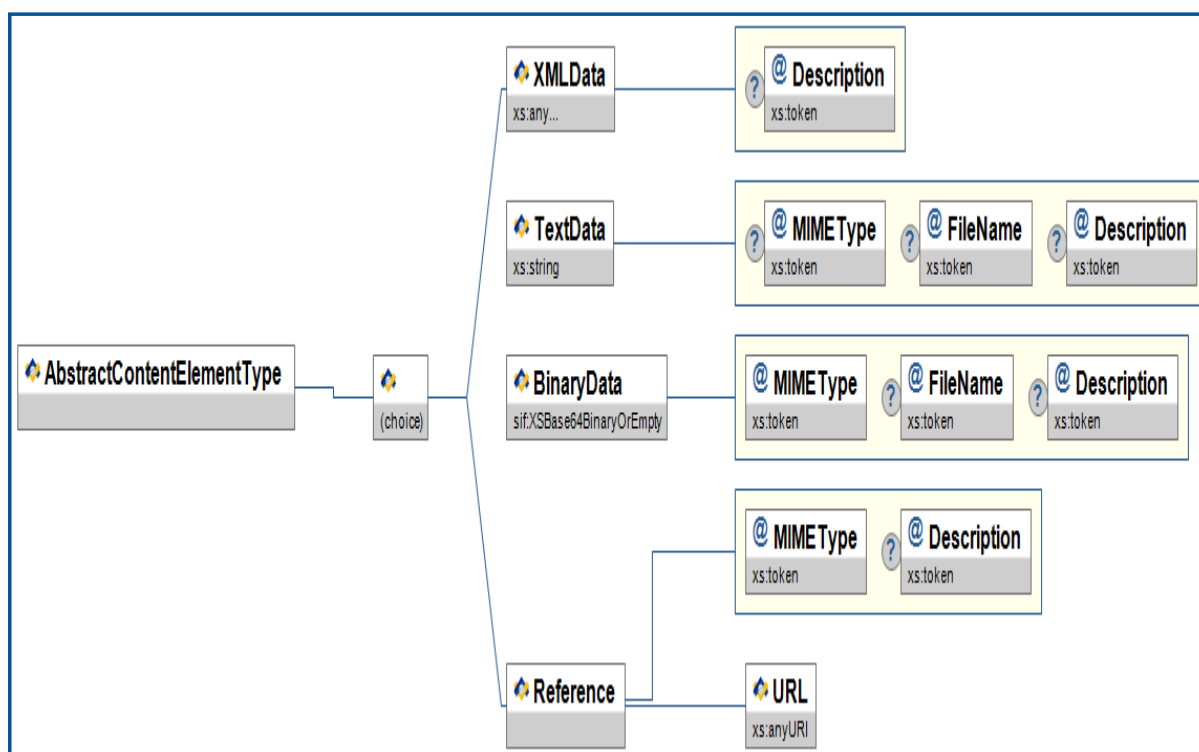


Figure A.1-1: AbstractContentType

Element/@ Attribute	Char	Description	Type
AbstractContentType		AbstractContentType used as an element rather than an object, omitting RefId , SIF_Metadata and SIF_ExtendedElements .	
XMLData	C	Contains an arbitrary XML element, encoded in UTF-8.	<xs:any processContents="lax" />
@ Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the AbstractContentType pattern.	xs:token

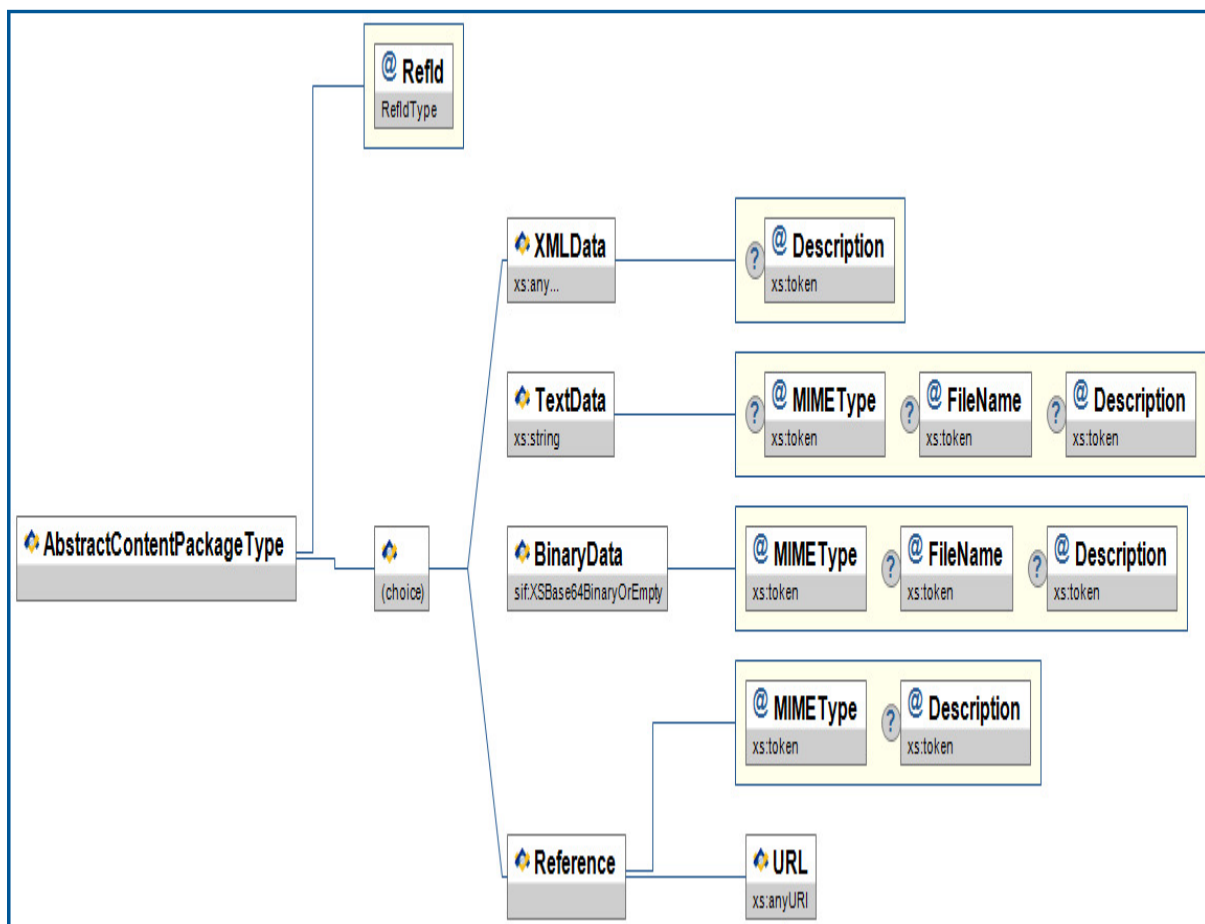
	Element/@ Attribute	Char	Description	Type
	TextData	C	Contains arbitrary text, encoded in UTF-8.	<code>xs:string</code>
@	MIMETYPE	O	Optional MIME type to specifically indicate the text type. Otherwise <code>text/plain</code> can be assumed.	<code>xs:token</code>
@	FileName	O	Optional file name to indicate the file from which the content originated, or to suggest a name to use when saving the content.	<code>xs:token</code>
@	Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the <code>AbstractContentPackageType</code> pattern.	<code>xs:token</code>
	BinaryData	C	Contains the <code>base64Binary</code> encoding of binary or text data not encoded in UTF-8.	<code>xs:base64Binary</code>
@	MIMETYPE	M	MIME type to indicate the content type.	<code>xs:token</code>
@	FileName	O	Optional file name to indicate the file from which the content originated, or to suggest a name to use when saving the content.	<code>xs:token</code>
@	Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the <code>AbstractContentPackageType</code> pattern.	<code>xs:token</code>
	Reference	C	References external content via a URL.	
@	MIMETYPE	M	MIME type to indicate the content type to be expected when retrieving the external content.	<code>xs:token</code>


Element/@Attribute	Char	Description	Type
@ Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the <code>AbstractContentPackageType</code> pattern.	<code>xs:token</code>
Reference/URL	M	Location of external content.	<code>xs:anyURI</code>

Table A.1-1: *AbstractContentElementType*

A.2 AbstractContentPackageType

An abstract type for derived content package types, elements and objects. This structure may be used verbatim, optionally extending with additional attributes, or may be redefined to include only a subset of child elements and/or to add validation to XML contained in `XMLData`. Only one instance of `XMLData`, `TextData`, `BinaryData` or `Reference` can occur in a single instance.

Figure A.2-1: *AbstractContentPackageType*

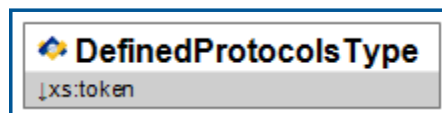
	Element/@Attribute	Char	Description	Type
	AbstractContentPackageType		An abstract type for derived content package types, elements and objects. This structure may be used verbatim, optionally extending with additional attributes, or may be redefined to include only a subset of child elements and/or to add validation to XML contained in <i>XMLData</i> . Only one instance of <i>XMLData</i> , <i>TextData</i> , <i>BinaryData</i> or <i>Reference</i> can occur in a single instance.	
@ 	RefId	M	The GUID that uniquely identifies an instance of the package.	RefIdType
	XMLData	C	Contains an arbitrary XML element, encoded in UTF-8.	<xs:any processContents="lax" />
@	Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the <i>AbstractContentPackageType</i> pattern.	xs:token
	TextData	C	Contains arbitrary text, encoded in UTF-8.	xs:string
@	MIMETYPE	O	Optional MIME type to specifically indicate the text type. Otherwise <i>text/plain</i> can be assumed.	xs:token
@	FileName	O	Optional file name to indicate the file from which the content originated, or to suggest a name to use when saving the content.	xs:token
@	Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the <i>AbstractContentPackageType</i> pattern.	xs:token
	BinaryData	C	Contains the base64Binary encoding of binary or text data not encoded in UTF-8.	xs:base64Binary

	Element/@Attribute	Char	Description	Type
@	MIMETYPE	M	MIME type to indicate the content type.	xs:token
@	FileName	O	Optional file name to indicate the file from which the content originated, or to suggest a name to use when saving the content.	xs:token
@	Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the <code>AbstractContentPackageType</code> pattern.	xs:token
	Reference	C	References external content via a URL.	
@	MIMETYPE	M	MIME type to indicate the content type to be expected when retrieving the external content.	xs:token
@	Description	O	Contains an optional description of the content or a processing hint with regard to its structure (e.g. named standard, file layout or XSD). Contents may be mandated in instances of this type, or types that follow the <code>AbstractContentPackageType</code> pattern.	xs:token
	Reference/URL	M	Location of external content.	xs:anyURI

Table A.2-1: *AbstractContentPackageType*

A.3 DefinedProtocolsType

The transport protocols defined in SIF.

Figure A.3-1: *DefinedProtocolsType*

	Element/@Attribute	Char	Description	Type
	DefinedProtocolsType		The transport protocols defined in SIF.	values: HTTPS HTTP

Table A.3-1: *DefinedProtocolsType*

A.4 ExtendedContentType

Allows for any mixed XML in an element.

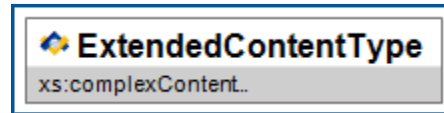


Figure A.4-1: ExtendedContentType

Element/@ Attribute	Char	Description	Type
ExtendedContentType		Allows for any mixed XML in an element.	<pre><xs:complexContent mixed="true"> <xs:restriction base="xs:anyType"> <xs:sequence> <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" /> </xs:sequence> </xs:restriction> </xs:complexContent></pre>

Table A.4-1: ExtendedContentType

A.5 GUIDType

SIF format for a GUID.



Figure A.5-1: GUIDType

Element/@ Attribute	Char	Description	Type
GUIDType		SIF format for a GUID.	<pre>xs:token xs:pattern [0-9A-F]{32}</pre>

Table A.5-1: GUIDType

A.6 IdRefType

A reference to a RefId.



Figure A.6-1: IdRefType

Element/@ Attribute	Char	Description	Type
IdRefType		A reference to a RefId.	<code>RefIdType</code>

Table A.6-1: IdRefType

A.7 MsgIdType

A message identifier.



Figure A.7-1: MsgIdType

Element/@ Attribute	Char	Description	Type
MsgIdType		A message identifier.	GUIDType

Table A.7-1: MsgIdType

A.8 ObjectNameType

An unenumerated SIF object name.



Figure A.8-1: ObjectNameType

Element/@ Attribute	Char	Description	Type
ObjectNameType		An unenumerated SIF object name.	xs:NCName xs:maxLength 64

Table A.8-1: ObjectNameType

A.9 ObjectType

A SIF XML object.

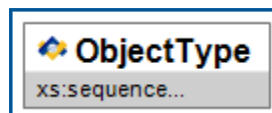


Figure A.9-1: ObjectType

Element/@ Attribute	Char	Description	Type
ObjectType		A SIF XML object.	<xs:sequence> <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" namespace /> </xs:sequence>

Table A.9-1: ObjectType

A.10 RefIdType

An object or element identifier.



Figure A.10-1: RefIdType

Element/@Attribute	Char	Description	Type
RefIdType		An object or element identifier.	GUIDType

Table A.10-1: RefIdType

A.11 ReportDataObjectType

A SIF XML object.



Figure A.11-1: ReportDataObjectType

Element/@Attribute	Char	Description	Type
ReportDataObjectType		A SIF XML object.	<pre><xs:sequence> <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" /> </xs:sequence></pre>

Table A.11-1: ReportDataObjectType

A.12 ReportPackageType

This package has exactly the same structure as AbstractContentPackageType. ReportPackage can be used in addition to SIF objects specifically in reporting situations within SIF_ReportObject. At this time, it is not a SIF object. It cannot be requested via SIF_Query or SIF_ExtendedQuery in a ReportManifest. It may be included in SIF_ReportObject as part of an external report definition.

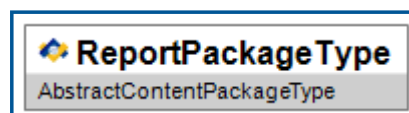


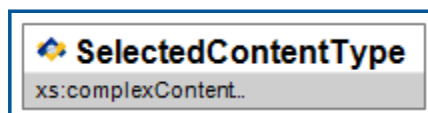
Figure A.12-1: ReportPackageType

Element/@Attribute	Char	Description	Type
ReportPackageType		This package has exactly the same structure as <code>AbstractContentPackageType</code> . <code>ReportPackage</code> can be used in addition to SIF objects specifically in reporting situations within <code>SIF_ReportObject</code> . At this time, it is not a SIF object. It cannot be requested via <code>SIF_Query</code> or <code>SIF_ExtendedQuery</code> in a <code>ReportManifest</code> . It may be included in <code>SIF_ReportObject</code> as part of an external report definition.	<code>AbstractContentPackageType</code>

Table A.12-1: *ReportPackageType*

A.13 SelectedContentType

Allows an XML fragment selected from an object to be used in an element with XML validation skipped.

Figure A.13-1: *SelectedContentType*

Element/@Attribute	Char	Description	Type
SelectedContentType		Allows an XML fragment selected from an object to be used in an element with XML validation skipped.	<pre><xs:complexContent mixed="true"> <xs:restriction base="xs:anyType"> <xs:sequence> <xs:any processContents="skip" minOccurs="0" maxOccurs="unbounded" /> </xs:sequence> </xs:restriction> </xs:complexContent></pre>

Table A.13-1: *SelectedContentType*

A.14 ServiceNameType

An unenumerated SIF object name.

Figure A.14-1: *ServiceNameType*

Element/@Attribute	Char	Description	Type
ServiceNameType		An unenumerated SIF object name.	<code>xs:NCName</code>

Table A.14-1: *ServiceNameType*

A.15 URIOrBinaryType

Allows for a URL or a Base-64 encoding.



Figure A.15-1: URIOrBinaryType

Element/@Attribute	Char	Description	Type
URIOrBinaryType		Allows for a URL or a Base-64 encoding.	union of: xs:anyURI xs:base64Binary

Table A.15-1: URIOrBinaryType

A.16 VersionType

A SIF version number.

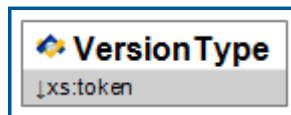


Figure A.16-1: VersionType

Element/@Attribute	Char	Description	Type				
VersionType		A SIF version number.	<div>xs:token</div> <table><tr><td>xs:pattern</td><td>[0-9]+[.][0-9]+(r[0-9]+)?</td></tr><tr><td>xs:maxLength</td><td>12</td></tr></table>	xs:pattern	[0-9]+[.][0-9]+(r[0-9]+)?	xs:maxLength	12
xs:pattern	[0-9]+[.][0-9]+(r[0-9]+)?						
xs:maxLength	12						

Table A.16-1: VersionType

A.17 VersionWithWildcardsType

A SIF version number, with wildcards for matching multiple versions.

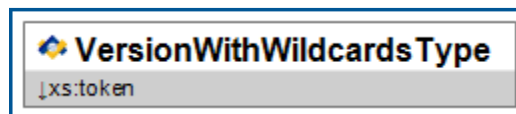


Figure A.17-1: VersionWithWildcardsType

Element/@ Attribute	Char	Description	Type				
VersionWithWildcardsType		A SIF version number, with wildcards for matching multiple versions.	<div>xs:token</div> <table><tr><td>xs:pattern</td><td>* ([0-9]+[.]*) ([0-9]+[.][0-9]+r*) ([0-9]+[.][0-9]+(r[0-9]+)?)</td></tr><tr><td>xs:maxLength</td><td>12</td></tr></table>	xs:pattern	* ([0-9]+[.]*) ([0-9]+[.][0-9]+r*) ([0-9]+[.][0-9]+(r[0-9]+)?)	xs:maxLength	12
xs:pattern	* ([0-9]+[.]*) ([0-9]+[.][0-9]+r*) ([0-9]+[.][0-9]+(r[0-9]+)?)						
xs:maxLength	12						

Table A.17-1: VersionWithWildcardsType

Appendix B: Code Sets

Select shared and named code sets defined in SIF are included here for reference.

Infrastructure

Status Code

0	Success (ZIS ONLY). SIF_Status/SIF_Data may contain additional data.
1	Immediate SIF_Ack (AGENT ONLY). Message is persisted or processing is complete. Discard the referenced message.
2	Intermediate SIF_Ack (AGENT ONLY). Only valid in response to SIF_Event delivery. Invokes Selective Message Blocking. The event referenced must still be persisted, and no other events must be delivered, until the agent sends a "Final" SIF_Ack at a later time.
3	Final SIF_Ack (AGENT ONLY). Sent (a SIF_Ack with this value is never returned by an agent in response to a delivered message) by an agent to the ZIS to end Selective Message Blocking. Discard the referenced event and allow for delivery of other events.
7	Already have a message with this SIF_MsgId from you.
8	Receiver is sleeping.
9	No messages available. This is returned when an agent is trying to pull messages from a ZIS and there are no messages available.

Error Category

The following table describes the functional areas where an error may occur in SIF. When a `SIF_Error` element is returned within a `SIF_Ack` message, the `SIF_Error/SIF_Category` element **MUST** contain one of the values from the table.

The next tables present the error codes that must be used when constructing a `SIF_Error` element. The value of `SIF_Error/SIF_Code` must come from these lists unless the functional category is `System` where error codes not defined in these tables can be included.

0	Unknown (This should NEVER be used if possible)
1	XML Validation
2	Encryption
3	Authentication
4	Access and Permissions
5	Registration
6	Provision
7	Subscription
8	Request and Response
9	Event Reporting and Processing
10	Transport
11	System (OS, Database, Vendor localized, etc.)
12	Generic Message Handling
13	SMB Handling
14	SIF Zone Service

XML Validation Error

1	Generic error
2	Message is not well-formed
3	Generic validation error
4	Invalid value for element/attribute
6	Missing mandatory element/attribute

Encryption Error

1	Generic error
---	---------------

Authentication Error

1	Generic error
2	Generic authentication error (with signature)
3	Missing sender's certificate
4	Invalid certificate
5	Sender's certificate is not trusted
6	Expired certificate
7	Invalid signature
8	Invalid encryption algorithm (only accepts MD4)
9	Missing public key of the receiver (when decrypting message)
10	Missing receiver's private key (when decrypting message)

Access and Permission Error

1	Generic error
2	No permission to register
3	No permission to provide this object
4	No permission to subscribe to this SIF_Event
5	No permission to request this object
6	No permission to respond to this object request
7	No permission to publish SIF_Event
8	No permission to administer policies
9	SIF_SourceId is not registered
10	No permission to publish SIF_Event Add
11	No permission to publish SIF_Event Change
12	No permission to publish SIF_Event Delete
13	No permission to publish indicated SIF_Notification
14	No permission to invoke SIF_ServiceInput to this Service
15	No permission to provide this Service

Registration Error

1	Generic error
---	---------------

2	The SIF_SourceId is invalid
3	Requested transport protocol is unsupported
4	Requested SIF_Version(s) not supported.
6	Requested SIF_MaxBufferSize is too small
7	ZIS requires a secure transport
9	Agent is registered for push mode (returned when a push-mode agent sends a SIF_GetMessage).
10	ZIS does not support the requested Accept-Encoding value.

Provision Error

1	Generic error
3	Invalid object
4	Object already has a provider (SIF_Provide message)

Subscription Error

1	Generic error
3	Invalid object

Request and Response Error

1	Generic error
3	Invalid object
4	No provider
7	Responder does not support requested SIF_Version
8	Responder does not support requested SIF_MaxBufferSize
9	Unsupported query in request
10	Invalid SIF_RequestMsgId specified in SIF_Response
11	SIF_Response is larger than requested SIF_MaxBufferSize
12	SIF_PacketNumber is invalid in SIF_Response
13	SIF_Response does not match any SIF_Version from SIF_Request
14	SIF_DestinationId does not match SIF_SourceId from SIF_Request
15	No support for SIF_ExtendedQuery
16	SIF_RequestMsgId deleted from cache due to timeout
17	SIF_RequestMsgId deleted from cache by administrator
18	SIF_Request cancelled by requesting agent
19	SIF_Request cancelled due to a SIF XML filter rule

Event Reporting and Processing Error

1	Generic error
3	Invalid event

Transport Error

1	Generic error
2	Requested protocol is not supported
3	Secure channel requested and no secure path exists
4	Unable to establish connection

System Error

1	Generic error
---	---------------

Generic Message Handling Error

1	Generic error
2	Message not supported
3	Version not supported
4	Context not supported
5	Protocol error
6	No such message (as identified by SIF_OriginalMsgId)
7	Multiple contexts not supported

SMB Error

1	Generic error
2	SMB can only be invoked during a SIF_Event acknowledgement
3	Final SIF_Ack expected from Push-Mode Agent
4	Incorrect SIF_MsgId in final SIF_Ack

SIF Zone Service Error

1	Generic error
2	Invalid service
3	No provider for service
4	Responder does not support requested SIF_Version
5	Responder does not support requested SIF_MaxBufferSize
6	Operation not supported
7	Argument not valid
8	Invalid SIF_ServiceMsgId specified in SIF_ServiceOutput
9	SIF_ServiceOutput is larger than requested SIF_MaxBufferSize
10	SIF_PacketNumber is invalid
11	SIF_ServiceOutput does not match any SIF_Version from SIF_ServiceInput
12	SIF_DestinationId does not match SIF_SourceId from SIF_ServiceInput
13	SIF_ServiceMsgId deleted from cache due to timeout
14	SIF_ServiceMsgId deleted from cache by administrator
15	SIF_ServiceInput cancelled by requesting agent
16	ACL permission denied

17	Not a provider for this service
18	Service or Operation failed

SIF_LogEntry

Agent Error Condition

1	An exception has occurred in the agent (generic error)
---	--

Data Issues with Failure Result

1	Insufficient information in message
2	Cannot process change due to business rule
3	Related information unavailable

Data Issues with Success Result

1	Data was changed to complete request successfully
2	Data was added to complete request successfully

Success Category

1	Success
---	---------

ZIS Error Condition

1	An exception has occurred in the ZIS (generic error)
2	Message could not be delivered due to buffer size limitations
3	Message could not be delivered due to minimum security requirements
4	Message could not be delivered due to destination agent not supporting SIF_Version
5	Message could not be delivered due to SIF_Response validation

Appendix C: Notes on Related Technologies

This partially normative appendix highlights technologies leveraged within SIF or related to SIF, either in their entirety or as a subset. It points out specifics casual readers of referenced documents on these technologies must not ignore when implementing SIF Zone Integration Servers and Agents.

C.1 SIF and HTTP(S)

SIF uses a small subset of HTTP 1.1 (SIF HTTP), as defined in

[Infrastructure Transport Layer](#), to promote interoperability. This section also defines a secure transport for SIF HTTP, SIF HTTPS, the required and default transport layer for use in SIF.

C.2 SIF and URLs

Zone Integration Servers and Push-mode Agents, when using SIF HTTPS or SIF HTTP, are addressable by an `http` or `https` Uniform Resource Locator (URL). As far as HTTP is concerned, these are simply formatted strings; no assumptions should be made about their format (e.g. that all ZIS URLs consist of a host, port and Zone Id, or that all agent URLs consist of a host, port and Agent Id) beyond the `http` and `https` schemes and the constituent parts from the generic URI (Uniform Resource Identifier) syntax

[\[RFC 2396\]](#).

```
http://host[:port][abs_path[?query]]
http://host[:port][abs_path[?query]]
```

Just because one Zone Integration Server seems to follow a certain convention with regard to its URLs, e.g.:

```
http://www.YourZIS.com/YourZone
```

does not imply another Zone Integration Server will not have a completely different format for a URL, for instance:

```
http://www.ZISesAreUs.com:8080/applications/ZIS;version=2.3.1?zone=ZoneA&cust=2A9823B2
```

or that a vendor's product might not change its URL conventions.

The same applies to URLs that address Push-mode Agents; conventions for URLs, within the general formatting that applies to URLs, can and do vary widely.

Zone Integration Servers and Agents **MUST** treat SIF HTTPS and SIF HTTP URLs as whole strings, whose only format rules stem from associated standards. This promotes interoperability as Zone administrators deploy Zone Integration Servers and Agents with different Zone configurations and products from different vendors.

C.3 SIF and XML

With its use in both Infrastructure and the SIF Data Model, SIF is greatly dependent on the structure and syntax of XML 1.0

[\[XML\]](#). SIF excludes the use of the `doctypeDecl` syntax from the optional `prolog` with which every XML document may begin. This implies that Zone Integration Servers and Agents **MUST NOT** reference an external DTD or internal DTD subset using the `doctypeDecl` production (e.g. `<!DOCTYPE SIF_Message ... !>`).

This should not be construed to imply that the rest of the XML `prolog` may not preface a SIF message, even though it never occurs in examples within this specification, being superfluous within SIF. As SIF mandates the use of XML 1.0, the character encoding of UTF-8 (contained in the HTTP `Content-Type` header), and all SIF messages are `standalone` due to the exclusion of `doctypeDecl` above, the values that can be communicated in the XML `prolog` are fixed within SIF. This implies that if a Zone Integration Server or Agent includes an XML `prolog` before a SIF message, it **MUST** take one of the following or equivalent forms (equivalent including case-insensitive character encoding names, XML's choice with regard to single or double quotes and optional spacing):

```
<?xml version="1.0"?>
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" standalone="yes"?>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

C.4 SIF and Unicode

The character set supported in XML 1.0 is Unicode/ISO 10646, a character set designed to be universal in nature with regard to its support for previously used character sets in the computer industry, ability to represent most human languages, numbers, commonly used symbols, etc. Thus the character set supported in SIF is Unicode/ISO 10646. If a Zone Integration Server or SIF-enabled application does not support Unicode/ISO 10646 internally, it **MUST** map Unicode/ISO 10646 to its local character set upon receipt of a SIF message and **MUST** map its local character set to Unicode/ISO 10646 when sending or responding to a SIF message. To promote interoperability and prevent loss of data in these conversions, it is **RECOMMENDED** that all Zone Integration Servers and SIF-enabled applications support Unicode/ISO 10646.

SIF HTTP further requires that the Unicode/ISO 10646 character set be encoded using the UTF-8 character encoding; Zone Integration Servers and Agents **MUST** encode SIF XML messages using UTF-8. To further promote interoperability, when the SIF Infrastructure or Data Model specifies that an octet/byte-based transformation of a text/string value be stored in a given element or attribute (e.g. Base64 encoding, hash value, encrypted form), Zone Integration Servers and Agents **MUST** convert the local character set of the value to Unicode/ISO 10646 if necessary, encode the resulting value using UTF-8, then apply the specified transformation.

C.5 SIF and XPath

SIF uses a small subset of XPath 1.0

[**XPATH**] in its own path syntax for referencing elements/attributes. This is defined in **SIF_Element Syntax**. This document may often use the same notation in referring to nested elements and/or attributes (e.g. `Name/FirstName`, `Name/@Type`), though it may include an object as the root element whereas the SIF_Element syntax does not (e.g. `StudentPersonal/Name/FirstName`, `StudentPersonal/@RefId`).

C.6 SIF and XML Schema

The SIF Association hosts and provides XML Schemas

[**SCHEMA**] for validating SIF messages, should Zone Integration Servers or Agents choose to perform message validation. These schemas leverage basic data types and structures as defined in that document. When these types and structures are referenced in this document they are prefixed with `xs:`.

Note that due to the ability of Zone Integration Servers and Agents to omit elements from data objects in the SIF Request/Response and SIF Event models, all elements defined as mandatory for SIF data objects in **Infrastructure** or **Data Model** and referenced common elements are defined as optional in the schema for validating any SIF_Message. The SIF Association hosts and provides alternate schemas that allow for validation of these data objects where mandatory elements cannot be omitted (e.g. in a `Add` event or in a `SIF_Response` where the `SIF_Request` did not specify a specific subset of elements to be returned from matching objects).

Notes on specific XML Schema types follow:

C.6.1 xs:boolean

Agents and Zone Integration servers **SHOULD** send values of `true` or `false`, but must understand equivalent `1` and `0` values.

C.6.2 xs:time

Agents and Zone Integration Servers **MUST** specify a time zone offset from UTC or indicate that the time is UTC unless the time zone is apparent locally from other elements/attributes per supplied documentation.

C.6.3 xs:date

Agents and Zone Integration Servers **MAY** specify a time zone offset or indicate UTC for dates, but in most cases do not need to do so unless zone activity spans great international distances.

C.6.4 xs:dateTime

Agents and Zone Integration Servers **MUST** specify a time zone offset from UTC or indicate that the time is UTC unless the time zone is apparent locally from other elements/attributes per supplied documentation.

Though use of a combined `xs:dateTime` may seem a natural fit for specifying a point in time, some SIF Association working groups and task forces prefer to separate `xs:dateTime` into element/attribute pairs of `xs:date` and `xs:time` per their object design/usage goals and/or for simplified querying. Applications wishing to query the date or time portion of `xs:dateTime` values may use comparison and boolean operators to do so.

C.7 SIF and XML Namespaces

Namespaces allow XML elements and attributes to be organized into units that allow for the separation of a set of names from others, effectively allowing the integration of XML defined from various sources to be included in the same XML document without risk of name/definition collisions. SIF has since its initial release used the default namespace attribute `xmlns`

[XMLNS] in the `SIF_Message` element. To a namespace-aware parser, the effective names of the elements in:

```
<SIF_Message Version="1.5r1" xmlns="http://www.sifinfo.org/infrastructure/1.x">
  <SIF_Event>...</SIF_Event>
</SIF_Message>
```

Example C.7-1: SIF_Message Namespace

are conceptually:

```
http://www.sifinfo.org/infrastructure/1.x:SIF_Message
http://www.sifinfo.org/infrastructure/1.x:SIF_Event
```

with the local names:

```
SIF_Message
SIF_Event
```

To a namespace-aware parser, the effective names of these same elements in the SIF 2.x namespace:

```
<SIF_Message Version="2.3" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Event>...</SIF_Event>
</SIF_Message>
```

Example C.7-2: SIF_Message Namespace

are conceptually:

```
http://www.sifinfo.org/infrastructure/2.x:SIF_Message
http://www.sifinfo.org/infrastructure/2.x:SIF_Event
```

with the local names:

```
SIF_Message
SIF_Event
```


A namespace-unaware parser simply interprets elements by their local names, and SIF 1.x and SIF 2.x elements are considered equivalent. If the local name is prefixed, a namespace-unaware parser considers the prefix and colon part of the name. To a namespace-unaware parser, `xml:lang` is named just that. To a namespace-aware parser, this is effectively `http://www.w3.org/XML/1998/namespace:lang` (the `xml` prefix is reserved in XML 1.0 and is always bound to this namespace in [XMLNS]) with a local name of `lang`.

Given the timing of the first release of SIF and the release of [Namespaces in XML \[XMLNS\]](#) it was never mandated in SIF that Zone Integration Servers and Agents be namespace-aware. Given the number of Zone Integration Servers and Agents that may at this point be namespace-unaware, it is not yet mandated that these components be namespace-aware, but this requirement may arise in a future major release of this specification. To allow for namespace-unaware parsers to reliably process SIF-defined XML by local names only, SIF messages **MUST** define the namespace for the corresponding SIF version as the default namespace of `SIF_Message` as documented in `SIF_Message`.

Furthermore, given the gradual proliferation of XML defined in other namespaces appearing in SIF XML, the following prefix-to-namespace mappings **MUST** be used should elements from these namespaces occur in SIF messages, to allow namespace-unaware parsers to reliably interpret names in these namespaces by local name:

Prefix	Namespace	Declaration
xml	<code>http://www.w3.org/XML/1998/namespace</code>	This is bound and fixed by default without declaration.
xsi	<code>http://www.w3.org/2001/XMLSchema-instance</code>	<code>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</code>
xs	<code>http://www.w3.org/2001/XMLSchema</code>	<code>xmlns:xs="http://www.w3.org/2001/XMLSchema"</code>

It is **RECOMMENDED** that other namespaces occurring in SIF messages (e.g. XML from outside SIF included in assessments, exchange of student records, etc.) have fixed prefix mappings, but it is not required. Affected elements **MAY** locally change the default namespace as desired, given that the default namespace for the `SIF_Message` as a whole remains the namespace for the corresponding SIF version.

When a fixed prefix is not defined for a given namespace, a namespace-unaware agent will be unable to reliably process these elements by name when prefixes vary, and must become namespace-aware to do so. XML not defined by SIF that in turn contains SIF-defined XML **MAY** reference SIF XML by its own prefix mapping rather than specifying the namespace of the corresponding SIF version as the default namespace using `xmlns`.

It is **RECOMMENDED** that as Zone Integration Servers and Agents are updated in their release schedules, they use namespace-aware parsers or parser options if they are not doing so already.

C.8 SIF and UUIDs/GUIDs

SIF leverages Universally Unique Identifiers (UUIDs), or Globally Unique Identifiers (GUIDs), as message and object identifiers, or primary keys, and occasionally for element identifiers internal to objects, per

[RFC 4122]. Note that SIF defines its own textual representation for GUIDs, uppercase and un-hyphenated (e.g. `F81D4FAE7DEC11D0A76500A0C91E6BF6` VS. `f81d4fae-7dec-11d0-a765-00a0c91e6bf6`). It should also be noted with SIF being a distributed system, to avoid the possibility of GUID collisions, especially in the SIF data model, systems generating GUIDs **SHOULD** use version 1 GUIDs which are unique in space as well as time when an IEEE 802 MAC address is available. Systems **MAY** use version 4 GUIDs which use a (pseudo-)random number-based algorithm if an IEEE 802 MAC address is unavailable or if the inclusion of that address in a GUID poses a compromising security risk.

C.9 SIF and Web Services

SIF is a web service, "a software system designed to support interoperable machine-to-machine interaction over a network

[WSARCH]. It is not a Web Service, as it lacks "an interface described in a machine-processable format (specifically WSDL) [WSARCH]." To meet this requirement and produce the Web Services Definition Language (WSDL) definition for SIF is a trivial exercise, creating a WSDL HTTP POST binding for the `SIF_Message-in/SIF_Message-out` exchange that describes the SIF HTTP(S) transport layer between Agents and ZIS, and between ZIS and Push-mode Agents. But the binding would be just that, a simple `SIF_Message-in/SIF_Message-out` exchange that doesn't capture the richness of the SIF infrastructure or necessarily provide the interoperability resulting from the precise definition of SIF HTTP(S). To do so and to meet the final requirement of a Web Service per [WSARCH], the use of SOAP messages, would require redefinition of much of SIF using SOAP messages. The SIF Association's Web Services Task Force has determined that this exercise has little value currently, given SIF's precisely defined

transport layer and installed base. The task force has left it as a future task how to best leverage Web Services in the future of SIF's infrastructure, if at all. In the meantime, the task force has, however, decided to provide a Web Services interface that provides external systems access to the rich amount of data available in SIF Zones via its own specification [[SIF Reporting WS](#)]. Future opportunities to provide additional services may be identified.

Appendix D: Wildcard Version Support Implementation Notes

Agents that register the ability to receive `SIF_Message`s defined by any number of different SIF Implementation Specification versions by using

version wildcards in `SIF_Register/SIF_Version` and `SIF_Request/SIF_Version` may receive messages defined by specification versions that did not exist at the time of agent implementation. This support can maximize agent communication in zones supporting multiple SIF versions; agent developers that design this support should be aware of the following implementation notes. These notes focus on wildcard support for releases **within a given major release lifecycle** and do not address agents that register support for `*`, indicating the ability to receive ANY version `SIF_Message`. These messages can be very different structurally across major version boundaries and an agent may require more sophisticated capabilities to successfully process any `SIF_Message`, regardless of the SIF version that defines it.

D.1 XML Parsing

The message handling protocols documented in this specification are written from the perspective of having a well-formed and—optionally—valid XML document and the ability to randomly access element and attribute values within the document in performing the message handling steps as documented. While some agent implementations have this ability, there do exist agent implementations that may process SIF XML using a streaming interface (e.g. SAX), processing an XML document node by node, to perform equivalent functionality. When these agents declare the ability to receive a `SIF_Message` defined by any minor release within a major release lifecycle, they cannot assume in processing a message that one element follows another without any intervening elements, as new minor releases of this specification can introduce optional elements into the SIF Data Model. An agent written at the time of SIF Implementation Specification 1.1 to support `1.*` and to expect `OtherId` to follow `AlertMsg` might encounter difficulties with processing a `1.5r1 StudentPersonal` if it were not designed to ignore new intervening `1.5r1` elements before `OtherId` unknown at the time of implementation, including `LocalId`, as shown here, not to mention `StatePrId` and `ElectronicId`, which were also both introduced in SIF Implementation Specification 1.5r1.

```
<StudentPersonal RefId="D3E34B359D75101A8C3D00AA001A1652">
  <AlertMessages>
    <AlertMessage Type="Legal">This is the Legal Alert for Joe Student</AlertMessage>
  </AlertMessages>
  <OtherIdList>
    <OtherId Type="0339">206654</OtherId>
  </OtherIdList>
  <Name Type="04">
    <LastName>Student</LastName>
    <FirstName>Joe</FirstName>
  </Name>
</StudentPersonal>
```

Example D.1-1: StudentPersonal from SIF Implementation Specification 1.1

```
<StudentPersonal RefId="D3E34B359D75101A8C3D00AA001A1652">
  <AlertMessages>
    <AlertMessage Type="Legal">Legal Alert for Joe Student</AlertMessage>
  </AlertMessages>
  <LocalId>P00001</LocalId>
  <OtherIdList>
    <OtherId Type="0339">206654</OtherId>
  </OtherIdList>
  <Name Type="04">
    <LastName>Student</LastName>
    <FirstName>Joe</FirstName>
  </Name>
</StudentPersonal>
```

Example D.1-2: StudentPersonal from SIF Implementation Specification 1.5r1

Agents that parse XML on a node-by-node basis and that wish to support wildcard versions must be able to read and skip XML elements not of interest until an expected element of interest is reached.

D.2 XML Validation

Though minor releases within a major version lifecycle of this specification are designed to be supersets of previous minor releases, agents supporting wildcard versions and performing XML validation should take into consideration that messages from a higher minor version in a major version lifecycle will not validate against schemas designed for a lower version, given the potential introduction of new objects, and new optional elements into existing data objects. Agents that do perform XML validation should skip validation of received `SIF_Message`s that are defined by a higher version, unless they have dynamic Internet access to hosted schemas where `SIF_Message/@Version` can be used to access schemas for new specification releases. These agents can, of course, still establish that received `SIF_Message` XML is well-formed and process that XML to access elements/attributes of interest to the agent implementation.

While `SIF_Message`s defined by lower minor versions in a major version lifecycle may validate against a higher-version schema in that lifecycle, it is recommended also that higher-version agents skip XML validation of lower-version `SIF_Message`s unless they have local access to schemas corresponding to the version in question, in which case the appropriate schema should be used for validation, or unless they have dynamic Internet access to hosted schemas where `SIF_Message/@Version` can be used to access schemas for other specification releases. This recommendation is made particularly because external code sets may be brought up to date with external sources with each release of this specification and a previously valid code set value may become invalid in a new specification.

Note that schemas hosted by the SIF Association are available at well-known URLs and can be used to dynamically access schemas for older/newer specification versions using `SIF_Message/@Version`, should agents with Internet access require them for XML validation:

`http://www.sifinfo.org/infrastructure/<value of SIF_Message/@Version>/DTD/SIF_Message.dtd` (for SIF 1.x—
`XSD/SIF_Message.xsd` also available)

`http://specification.sifinfo.org/Implementation/<value of SIF_Message/@Version>/XSD/SIF_Message.xsd` (for SIF 2.x)

D.3 SIF_Message Handling

While this is defined in the

[SIF_Message Agent Message Handling Protocol](#), it bears repeating in this section that agents receiving an unexpected message from the ZIS respond according to protocol, acknowledging receipt of the message with a `SIF_Ack` including the `SIF_Error` element with a `SIF_Category` of 12 (Generic Message Handling) and a `SIF_Code` of 2 (message not supported). This allows an agent with wildcard version support to successfully ignore `SIF_Message`s that may be introduced with the addition of optional infrastructure functionality into new minor releases of this specification, including new `SIF_SystemControl` messages.

Appendix E: Selective Message Blocking (SMB) Example

Note: SMB functionality does not apply to Zone Services. Notification messages will not be blocked.

E.1 Example

A detailed example of Selective Message Blocking (SMB) follows. The table below represents the agent's message queue as maintained by the ZIS. The message at the top represents the oldest message in the queue and is the message that is currently being processed by the agent as the example begins.

Agent Message Queue
SIF_Event message containing a StudentSchoolEnrollment object with an Action of Add.
SIF_Event message containing a StudentPersonal object with an Action of Add.
SIF_Request message for a StudentPersonal object from another agent.
SIF_Event message containing a StudentSchoolEnrollment object with an Action of Add.

Table E.1-1: Agent Message Queue - Example 1

When processing the StudentSchoolEnrollment event, the agent requires data from a SchoolInfo object that it doesn't have locally. It would like to request the SchoolInfo object without needing to process subsequent events. To do so, the agent acknowledges the StudentSchoolEnrollment event with an "Intermediate" SIF_Ack indicating that the ZIS will be contacted later to resume delivery of events. It then opens a channel to the ZIS and submits a SIF_Request for the SchoolInfo object.

Upon receipt of the "Intermediate" SIF_Ack, the ZIS freezes the delivery of any SIF_Event messages to this agent until the agent sends a final SIF_Ack releasing the original event. The current state of the queue is now:

Agent Message Queue
SIF_Event message containing a StudentSchoolEnrollment object with an Action of Add. (blocked)
SIF_Event message containing a StudentPersonal object with an Action of Add. (frozen)
SIF_Request message for a StudentPersonal object from another agent.
SIF_Event message containing a StudentSchoolEnrollment object with an Action of Add. (frozen)

Table E.1-2: Agent Message Queue - Example 2

The next message available for delivery to the agent is the SIF_Request for a StudentPersonal object. For our example, the agent will accept the SIF_Request by returning an "Immediate" SIF_Ack indicating that processing is complete and the agent will hand the SIF_Request off to another part of the agent for handling.

Meanwhile, the ZIS has deposited the SIF_Response from the SchoolInfo provider's agent into the queue. The queue now looks like this:

Agent Message Queue
SIF_Event message containing a StudentSchoolEnrollment object with an Action of Add. (blocked)
SIF_Event message containing a StudentPersonal object with an Action of Add. (frozen)
SIF_Event message containing a StudentSchoolEnrollment object with an Action of Add. (frozen)
SIF_Response message containing the SchoolInfo object previously requested.

Table E.1-3: Agent Message Queue - Example 3

The next message the agent receives is the SIF_Response. The agent takes the SIF_Response and uses the information from it along with the data in the original StudentSchoolEnrollment event to update its database. The agent returns (Pull-Mode) or sends (Push-Mode) an "Immediate" SIF_Ack telling the ZIS to discard the SIF_Response message.

The agent has now completed processing of the StudentSchoolEnrollment event and opens a channel to the ZIS and sends a "Final" SIF_Ack with the message identifier for the StudentSchoolEnrollment event. The SIF_Ack says that the agent has completed

processing and the ZIS removes the event from the agent queue. The freeze on `SIF_Event` messages is lifted and the next message to be sent to the agent is the `SIF_Event` for a `StudentPersonal Add`:

Agent Message Queue
<code>SIF_Event</code> message containing a <code>StudentPersonal</code> object with an Action of <code>Add</code> .
<code>SIF_Event</code> message containing a <code>StudentSchoolEnrollment</code> object with an Action of <code>Add</code> .

Table E.1-4: Agent Message Queue - Example 4

"Immediate" `SIF_Ack`

The "Immediate" `SIF_Ack` is a `SIF_Ack` message with status code of 1. This type of `SIF_Ack` is returned as a response to a message sent by the ZIS and indicates that the agent has persisted or has processed the message and the ZIS must remove the message from its queue.

```
<SIF_Message Version="2.3" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>ABCD10580EF250789012AC0554321EA2</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>10580EF2ABCD50789012AC05EA6C71B3</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>1</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example E.1-1: "Immediate" `SIF_Ack`

"Intermediate" `SIF_Ack`

The "Intermediate" `SIF_Ack` is a `SIF_Ack` message with status code of 2. This type of `SIF_Ack` is returned as a response to an event message delivered by the ZIS and indicates that the agent has not completed processing of the event and the ZIS must not remove the event message from its queue. The agent will send a "Final" `SIF_Ack` to the ZIS in the future to signal that the ZIS can discard the event message. An "Intermediate" `SIF_Ack` message must not be returned by agents in response to messages other than `SIF_Event`.

```
<SIF_Message Version="2.3" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>ABCD10580EF250789012AC0554321EA3</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>10580EF2ABCD50789012AC05EA6C71B3</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>2</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example E.1-2: "Intermediate" `SIF_Ack`

"Final" `SIF_Ack`

A "Final" `SIF_Ack` is a message with status code of 3. The agent sends this type of `SIF_Ack` to the ZIS after the agent has completely processed a `SIF_Event` where it previously sent an "Intermediate" `SIF_Ack`. When the ZIS receives this message, it must discard the `SIF_Event` message referenced in the `SIF_Ack` upon successfully acknowledging the "Final" `SIF_Ack`.

```
<SIF_Message Version="2.3" xmlns="http://www.sifinfo.org/infrastructure/2.x">
  <SIF_Ack>
    <SIF_Header>
      <SIF_MsgId>ABCD10580EF250789012AC0554321EA4</SIF_MsgId>
      <SIF_Timestamp>2006-02-18T08:39:40-08:00</SIF_Timestamp>
      <SIF_SourceId>RamseyLIB</SIF_SourceId>
    </SIF_Header>
    <SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
    <SIF_OriginalMsgId>10580EF2ABCD50789012AC05EA6C71B3</SIF_OriginalMsgId>
    <SIF_Status>
      <SIF_Code>3</SIF_Code>
    </SIF_Status>
  </SIF_Ack>
</SIF_Message>
```

Example E.1-3: "Final" SIF_Ack

Appendix F: Index of Tables

Table 3.3.8-1	Differences between a Data Object and a Zone Service
Table 3.5.1-1	Register
Table 3.5.1-2	Virtual Table Example (Register)
Table 3.5.1-3	Access Control
Table 3.5.1-4	Virtual Table Example (Access Control)
Table 3.5.3-1	XML Filter Example 1
Table 3.5.3-2	XML Filter Example SIF_LogEntry
Table 3.6.3.4-1	Key Lengths
Table 3.7.1.3-1	HTTP Request Headers
Table 3.7.1.4-1	HTTP Response Headers
Table 4.1.1.1-1	SIF_Register Protocol
Table 4.1.1.2-1	SIF_Unregister Protocol
Table 4.1.1.3-1	SIF_Provide Protocol
Table 4.1.1.4-1	SIF_Unprovide Protocol
Table 4.1.1.5-1	SIF_Subscribe Protocol
Table 4.1.1.6-1	SIF_Unsubscribe Protocol
Table 4.1.1.7-1	SIF_Provision Protocol
Table 4.1.1.8-1	SIF_Event Protocol
Table 4.1.1.9-1	SIF_Request Protocol
Table 4.1.1.10-1	SIF_Ping Protocol
Table 4.1.1.11-1	SIF_Sleep Protocol
Table 4.1.1.12-1	SIF_Wakeup Protocol
Table 4.1.1.13-1	SIF_GetZoneStatus Protocol
Table 4.1.1.14-1	SIF_GetAgentACL Protocol
Table 4.1.1.15-1	SIF_CancelRequests Protocol
Table 4.1.1.16-1	SIF_GetMessage Protocol
Table 4.1.1.17-1	SIF_Ack Protocol (Push-Mode)
Table 4.1.1.18-1	SIF_Ack Protocol (Pull-Mode)
Table 4.1.1.19-1	SIF_ServiceNotify Protocol
Table 4.1.1.20-1	SIF_ServiceInput Protocol
Table 4.1.1.21-1	SIF_CancelServiceInputs Protocol
Table 4.1.2.1-1	SIF_Message Handling
Table 4.1.2.2-1	SIF_Event Handling
Table 4.1.2.3-1	SIF_Request Handling
Table 4.1.2.4-1	SIF_Response Handling
Table 4.1.2.5-1	SIF_Ping Handling
Table 4.1.2.6-1	SIF_Sleep Handling
Table 4.1.2.7-1	SIF_Wakeup Handling
Table 4.1.2.8-1	SIF_CancelRequests Handling
Table 4.1.2.9-1	SIF_CancelServiceInputs Handling
Table 4.1.2.10-1	SIF_ServiceNotify Handling
Table 4.1.2.11-1	SIF_ServiceInput Handling
Table 4.1.2.12-1	SIF_ServiceOutput Handling
Table 4.2.1.1-1	SIF_Message Delivery Protocol
Table 4.2.1.2-1	SIF_Ping Protocol
Table 4.2.1.3-1	SIF_Sleep Protocol
Table 4.2.1.4-1	SIF_Wakeup Protocol
Table 4.2.1.5-1	SIF_CancelRequests Protocol
Table 4.2.1.6-1	SIF_CancelServiceInputs Protocol

Table 4.2.2.1-1	SIF_Message Handling
Table 4.2.2.2-1	SIF_Register Handling
Table 4.2.2.3-1	SIF_Unregister Handling
Table 4.2.2.4-1	SIF_Provide Handling
Table 4.2.2.5-1	SIF_Unprovide Handling
Table 4.2.2.6-1	SIF_Subscribe Handling
Table 4.2.2.7-1	SIF_Unsubscribe Handling
Table 4.2.2.8-1	SIF_Provision Handling
Table 4.2.2.9-1	SIF_Event Handling
Table 4.2.2.10-1	SIF_Request Handling
Table 4.2.2.11-1	SIF_Response Handling
Table 4.2.2.12-1	SIF_Ping Handling
Table 4.2.2.13-1	SIF_Sleep Handling
Table 4.2.2.14-1	SIF_Wakeup Handling
Table 4.2.2.15-1	SIF_GetZoneStatus Handling
Table 4.2.2.16-1	SIF_GetZoneStatus Handling
Table 4.2.2.17-1	SIF_CancelRequests Handling
Table 4.2.2.18-1	SIF_CancelRequests Handling
Table 4.2.2.19-1	SIF_GetMessage Handling
Table 4.2.2.20-1	SIF_Ack Handling
Table 4.2.2.21-1	SIF_Ack Handling
Table 4.2.2.22-1	SIF_ServiceInput Handling
Table 4.2.2.23-1	SIF_ServiceInput Handling
Table 4.2.2.24-1	SIF_ServiceOutput Handling
Table 5.1.1-1	SIF_ExtendedElements
Table 5.1.2-1	SIF_Message
Table 5.1.3-1	SIF_Header
Table 5.1.4-1	SIF_EncryptionLevel
Table 5.1.5-1	SIF_AuthenticationLevel
Table 5.1.6-1	SIF_Contexts
Table 5.1.7-1	SIF_Context
Table 5.1.8-1	SIF_Protocol
Table 5.1.9-1	SIF_Status
Table 5.1.10-1	SIF_Error
Table 5.1.11-1	SIF_Query
Table 5.1.12-1	SIF_ExtendedQuery
Table 5.1.12.1-1	Mapping SIF_Query to SIF_ExtendedQuery
Table 5.1.13-1	SIF_ExtendedQueryResults
Table 5.2.1-1	SIF_Ack
Table 5.2.2-1	SIF_Event
Table 5.2.3-1	SIF_Provide
Table 5.2.4-1	SIF_Provision
Table 5.2.5-1	SIF_Register
Table 5.2.6-1	SIF_Request
Table 5.2.7-1	SIF_Response
Table 5.2.8-1	SIF_Subscribe
Table 5.2.9-1	SIF_SystemControl
Table 5.2.10-1	SIF_Ping
Table 5.2.11-1	SIF_Sleep
Table 5.2.12-1	SIF_Wakeup
Table 5.2.13-1	SIF_GetMessage
Table 5.2.14-1	SIF_GetZoneStatus

Table 5.2.15-1	SIF_GetAgentACL
Table 5.2.16-1	SIF_CancelRequests
Table 5.2.17-1	SIF_CancelServiceInputs
Table 5.2.18-1	SIF_Unprovide
Table 5.2.19-1	SIF_Unregister
Table 5.2.20-1	SIF_Unsubscribe
Table 5.2.21-1	SIF_ServiceInput
Table 5.2.22-1	SIF_ServiceOutput
Table 5.2.23-1	SIF_ServiceNotify
Table 5.3.1-1	SIF_AgentACL
Table 5.3.2-1	SIF_LogEntry
Table 5.3.3-1	SIF_ZoneStatus
Table A.1-1	AbstractContentElementType
Table A.2-1	AbstractContentPackageType
Table A.3-1	DefinedProtocolsType
Table A.4-1	ExtendedContentType
Table A.5-1	GUIDType
Table A.6-1	IdRefType
Table A.7-1	MsgIdType
Table A.8-1	ObjectNameType
Table A.9-1	ObjectType
Table A.10-1	RefIdType
Table A.11-1	ReportDataObjectType
Table A.12-1	ReportPackageType
Table A.13-1	SelectedContentType
Table A.14-1	ServiceNameType
Table A.15-1	URIOrBinaryType
Table A.16-1	VersionType
Table A.17-1	VersionWithWildcardsType
Table E.1-1	Agent Message Queue - Example 1
Table E.1-2	Agent Message Queue - Example 2
Table E.1-3	Agent Message Queue - Example 3
Table E.1-4	Agent Message Queue - Example 4

Appendix G: Index of Examples

Example 2.2.3-1	Examples Convention
Example 3.6.6.2-1	The "Pull" Model - SIF_Status/SIF_Code of 0
Example 3.6.6.2-2	The "Pull" Model - SIF_Status/SIF_Code of 9
Example 3.7.1.3-1	SIF HTTPS Request
Example 3.7.1.4-1	SIF HTTPS Response
Example 3.7.3-1	SIF client requesting compression of response
Example 3.7.3-2	SIF server returning compressed SIF_Ack
Example 3.7.3-3	SIF client sending compressed SIF_Message
Example 3.7.3-4	SIF client sending compressed SIF_Message and requesting compression of response
Example 3.7.4-1	SIF_Protocol with Accept-Encoding indicating acceptance of gzip (and identity)
Example 3.7.4-2	SIF_Protocol with Accept-Encoding indicating no acceptance of encodings other than gzip or identity, gzip preferred over identity
Example 5.1.1-1	SIF_ExtendedElements
Example 5.1.2-1	SIF_Message
Example 5.1.3-1	SIF_Header
Example 5.1.3-2	SIF_Header
Example 5.1.11.1-1	
Example 5.1.11.1-2	
Example 5.1.11.1-3	SIF_ConditionGroup querying into an object
Example 5.1.11.2-1	
Example 5.1.11.2-2	
Example 5.1.12-1	Selecting all StudentPersonal objects
Example 5.1.12-3	Selecting all attributes and immediate child elements of StudentPersonal as columns from all StudentPersonal objects
Example 5.1.12-5	Selecting specific attributes and elements from all StudentPersonal objects
Example 5.1.12-7	Selecting StudentPersonal objects along with each student's EntryDate from StudentSchoolEnrollment for a specific school, school year and other StudentSchoolEnrollment values, sorted by student's last name
Example 5.1.12-9	Selecting a specific StudentPersonal's StudentSchoolEnrollment objects, along with the corresponding school name for each enrollment
Example 5.1.12.1-1	Input SIF_Query
Example 5.1.12.1-2	Corresponding SIF_ExtendedQuery
Example 5.1.13-1	SIF_ExtendedQueryResults
Example 5.2.1-1	SIF_Ack Status Message
Example 5.2.1-4	SIF_Ack Error Message
Example 5.2.2-1	SIF_Event Message with StudentPersonal changes
Example 5.2.3-1	SIF_Provide
Example 5.2.4-1	SIF_Provision
Example 5.2.5-1	SIF_Register
Example 5.2.6-1	SIF_Request
Example 5.2.7-1	Sample single-packet SIF_Response to a SIF_Request for the Name element from a StudentPersonal object
Example 5.2.7-3	SIF_Response (first packet)
Example 5.2.7-5	SIF_Response (second packet)
Example 5.2.7-7	SIF_Response with no matching objects
Example 5.2.8-1	SIF_Subscribe
Example 5.2.9-1	SIF_SystemControl
Example 5.2.10-1	SIF_Ping
Example 5.2.10-3	SIF_SystemControl—SIF_Ping ("Okay" status)
Example 5.2.10-5	SIF_SystemControl—SIF_Ping ("Receiver is sleeping" status)
Example 5.2.10-7	SIF_SystemControl—SIF_Ping (Transport error returned)
Example 5.2.11-1	SIF_Sleep
Example 5.2.11-3	SIF_Ack with "Okay" status in response to SIF_Sleep
Example 5.2.12-1	SIF_Wakeup

Example 5.2.12-3	SIF_Ack with an "Okay" status in response to SIF_Wakeup
Example 5.2.13-1	SIF_GetMessage
Example 5.2.13-3	SIF_Ack in response to SIF_GetMessage
Example 5.2.13-5	SIF_Ack in response to SIF_GetMessage (no message in queue)
Example 5.2.14-1	SIF_GetZoneStatus
Example 5.2.14-3	SIF_Ack containing SIF_ZoneStatus
Example 5.2.16-1	SIF_CancelRequests
Example 5.2.17-1	SIF_CancelServiceInputs
Example 5.2.18-1	SIF_Unprovide
Example 5.2.19-1	SIF_Unregister
Example 5.2.20-1	SIF_Unsubscribe
Example 5.2.21-1	Example 1 - Simple SIF_ServiceInput
Example 5.2.22-1	SIF_ServiceOutput
Example 5.2.23-1	SIF_ServiceNotify
Example 5.3.1-1	SIF_AgentACL
Example 5.3.2-1	SIF_LogEntry when an agent encounters a system failure
Example 5.3.2-2	SIF_LogEntry when an agent fails to delete a student
Example 5.3.2-3	SIF_LogEntry when an agent starts synchronizing data
Example 5.3.2-4	SIF_LogEntry when a ZIS fails to deliver a message due to buffer size limitations
Example 5.3.3-1	SIF_ZoneStatus
Example C.7-1	SIF_Message Namespace
Example C.7-2	SIF_Message Namespace
Example D.1-1	StudentPersonal from SIF Implementation Specification 1.1
Example D.1-2	StudentPersonal from SIF Implementation Specification 1.5r1
Example E.1-1	"Immediate" SIF_Ack
Example E.1-2	"Intermediate" SIF_Ack
Example E.1-3	"Final" SIF_Ack

Appendix H: Index of Figures

Figure 2.2.6-1	XML Diagram Conventions
Figure 3.2-1	Single-Zone School SIF Implementation
Figure 3.2.1-1	Multiple-Zone District SIF Implementation
Figure 3.2.1-2	Multiple-Zone State SIF Implementation
Figure 3.3.2-1	Zone Architecture Block Diagram
Figure 3.3.3-1	Publish/Subscribe Model
Figure 3.3.3-2	Request/Response Model
Figure 3.3.3-3	Message Structure
Figure 3.3.5-1	Subscribe to Events
Figure 3.3.5-2	Register with ZIS
Figure 3.3.7-1	Security Model
Figure 3.4.2-1	SIF Event
Figure 4.1.1.1-1	SIF_Register Agent Message Protocol
Figure 4.1.1.2-1	SIF_Unregister Agent Message Protocol
Figure 4.1.1.3-1	SIF_Provide Agent Message Protocol
Figure 4.1.1.4-1	SIF_Unprovide Agent Message Protocol
Figure 4.1.1.5-1	SIF_Subscribe Agent Message Protocol
Figure 4.1.1.6-1	SIF_Unsubscribe Agent Message Protocol
Figure 4.1.1.7-1	SIF_Provision Agent Message Protocol
Figure 4.1.1.8-1	SIF_Event Agent Message Protocol
Figure 4.1.1.9-1	SIF_Request Agent Message Protocol
Figure 4.1.1.10-1	SIF_Ping Agent Message Protocol
Figure 4.1.1.11-1	SIF_Sleep Agent Message Protocol
Figure 4.1.1.12-1	SIF_Wakeup Agent Message Protocol
Figure 4.1.1.13-1	SIF_GetZoneStatus Agent Message Protocol
Figure 4.1.1.14-1	SIF_GetAgentACL Agent Message Protocol
Figure 4.1.1.15-1	SIF_CancelRequests Agent Message Protocol
Figure 4.1.1.16-1	SIF_GetMessage (Pull-Mode only) Agent Message Protocol
Figure 4.1.1.17-1	SIF_Ack (Push-Mode) Agent Message Protocol
Figure 4.1.1.18-1	SIF_Ack (Pull-Mode) Agent Message Protocol
Figure 4.1.1.19-1	SIF_ServiceNotify Agent Message Protocol
Figure 4.1.1.20-1	SIF_ServiceInput Agent Message Protocol
Figure 4.1.1.21-1	SIF_CancelServiceInputs Agent Message Protocol
Figure 5.1.1-1	SIF_ExtendedElements
Figure 5.1.2-1	SIF_Message
Figure 5.1.3-1	SIF_Header
Figure 5.1.4-1	SIF_EncryptionLevel
Figure 5.1.5-1	SIF_AuthenticationLevel
Figure 5.1.6-1	SIF_Contexts
Figure 5.1.7-1	SIF_Context
Figure 5.1.8-1	SIF_Protocol
Figure 5.1.9-1	SIF_Status
Figure 5.1.10-1	SIF_Error
Figure 5.1.11-1	SIF_Query
Figure 5.1.12-1	SIF_ExtendedQuery
Figure 5.1.13-1	SIF_ExtendedQueryResults
Figure 5.2.1-1	SIF_Ack
Figure 5.2.2-1	SIF_Event
Figure 5.2.3-1	SIF_Provide

Figure 5.2.4-1	SIF_Provision
Figure 5.2.5-1	SIF_Register
Figure 5.2.6-1	SIF_Request
Figure 5.2.7-1	SIF_Response
Figure 5.2.8-1	SIF_Subscribe
Figure 5.2.9-1	SIF_SystemControl
Figure 5.2.10-1	SIF_Ping
Figure 5.2.11-1	SIF_Sleep
Figure 5.2.12-1	SIF_Wakeup
Figure 5.2.13-1	SIF_GetMessage
Figure 5.2.14-1	SIF_GetZoneStatus
Figure 5.2.15-1	SIF_GetAgentACL
Figure 5.2.16-1	SIF_CancelRequests
Figure 5.2.17-1	SIF_CancelServiceInputs
Figure 5.2.18-1	SIF_Unprovide
Figure 5.2.19-1	SIF_Unregister
Figure 5.2.20-1	SIF_Unsubscribe
Figure 5.2.21-1	SIF_ServiceInput
Figure 5.2.22-1	SIF_ServiceOutput
Figure 5.2.23-1	SIF_ServiceNotify
Figure 5.3.1-1	SIF_AgentACL
Figure 5.3.2-1	SIF_LogEntry
Figure 5.3.3-1	SIF_ZoneStatus
Figure A.1-1	AbstractContentElementType
Figure A.2-1	AbstractContentPackageType
Figure A.3-1	DefinedProtocolsType
Figure A.4-1	ExtendedContentType
Figure A.5-1	GUIDType
Figure A.6-1	IdRefType
Figure A.7-1	MsgIdType
Figure A.8-1	ObjectNameType
Figure A.9-1	ObjectType
Figure A.10-1	RefIdType
Figure A.11-1	ReportDataObjectType
Figure A.12-1	ReportPackageType
Figure A.13-1	SelectedContentType
Figure A.14-1	ServiceNameType
Figure A.15-1	URIOrBinaryType
Figure A.16-1	VersionType
Figure A.17-1	VersionWithWildcardsType

Appendix I: Index of Objects

SIF_AgentACL	5.3.1
SIF_LogEntry	5.3.2
SIF_ZoneStatus	5.3.3

3 Total

Appendix J: Index of Common Elements

SIF_AuthenticationLevel	5.1.5
SIF_Context	5.1.7
SIF_Contexts	5.1.6
SIF_EncryptionLevel	5.1.4
SIF_Error	5.1.10
SIF_ExtendedElements	5.1.1
SIF_ExtendedQuery	5.1.12
SIF_ExtendedQueryResults	5.1.13
SIF_Header	5.1.3
SIF_Message	5.1.2
SIF_Protocol	5.1.8
SIF_Query	5.1.11
SIF_Status	5.1.9

13 Total

Appendix K: Index of Common Types

AbstractContentElementType	A.1
AbstractContentPackageType	A.2
DefinedProtocolsType	A.3
ExtendedContentType	A.4
GUIDType	A.5
IdRefType	A.6
MsgIdType	A.7
ObjectNameType	A.8
ObjectType	A.9
RefIdType	A.10
ReportDataObjectType	A.11
ReportPackageType	A.12
SelectedContentType	A.13
ServiceNameType	A.14
URIOrBinaryType	A.15
VersionType	A.16
VersionWithWildcardsType	A.17

17 Total

Appendix L: Index of Elements

AbstractContentType	A.1-1.1
AbstractContentPackageType	A.2-1.1
BinaryData	A.1-1.8, A.2-1.9
C	5.1.13-1.9
DefinedProtocolsType	A.3-1.1
ExtendedContentType	A.4-1.1
GUIDType	A.5-1.1
IdRefType	A.6-1.1
MsgIdType	A.7-1.1
ObjectNameType	A.8-1.1
ObjectType	A.9-1.1
R	5.1.13-1.8
Reference	A.1-1.12, A.2-1.13
RefIdType	A.10-1.1
ReportDataObjectType	A.11-1.1
ReportPackageType	A.12-1.1
SelectedContentType	A.13-1.1
ServiceNameType	A.14-1.1
SIF_Ack	5.2.1-1.1
SIF_AddPublishers	5.3.3-1.24
SIF_AdministrationURL	5.3.3-1.87
SIF_AgentACL	5.3.1-1.1
SIF_Application	5.2.5-1.10, 5.3.3-1.68
SIF_ApplicationCode	5.3.2-1.10
SIF_AuthenticationLevel	5.1.3-1.6, 5.1.5-1.1, 5.3.3-1.77
SIF_Body	5.2.21-1.11, 5.2.22-1.7, 5.2.23-1.9
SIF_CancelRequests	5.2.16-1.1
SIF_CancelServiceInputs	5.2.17-1.1
SIF_Category	5.1.10-1.2, 5.3.2-1.8
SIF_ChangePublishers	5.3.3-1.31
SIF_Code	5.1.9-1.2, 5.1.10-1.3, 5.3.2-1.9
SIF_ColumnHeaders	5.1.13-1.2
SIF_Condition	5.1.11-1.9, 5.1.12-1.23
SIF_ConditionGroup	5.1.11-1.5, 5.1.12-1.19
SIF_Conditions	5.1.11-1.7, 5.1.12-1.21
SIF_Context	5.1.6-1.2, 5.1.7-1.1
SIF_Contexts	5.1.3-1.10, 5.1.6-1.1, 5.2.3-1.6, 5.2.4-1.7, 5.2.4-1.11, 5.2.4-1.15, 5.2.4-1.19, 5.2.4-1.23, 5.2.4-1.28, 5.2.4-1.33, 5.2.4-1.37, 5.2.4-1.41, 5.2.4-1.45, 5.2.4-1.51, 5.2.8-1.5, 5.2.18-1.5, 5.2.20-1.5, 5.3.1-1.5, 5.3.1-1.9, 5.3.1-1.13, 5.3.1-1.17, 5.3.1-1.21, 5.3.1-1.25, 5.3.1-1.29, 5.3.1-1.33, 5.3.1-1.37, 5.3.1-1.41, 5.3.1-1.47, 5.3.3-1.16, 5.3.3-1.23, 5.3.3-1.30, 5.3.3-1.37, 5.3.3-1.44, 5.3.3-1.52, 5.3.3-1.60, 5.3.3-1.88, 5.3.3-1.95, 5.3.3-1.102, 5.3.3-1.111, 5.3.3-1.120
SIF_Data	5.1.9-1.4
SIF_DeletePublishers	5.3.3-1.38
SIF_Desc	5.1.9-1.3, 5.1.10-1.4, 5.3.2-1.11
SIF_DestinationId	5.1.3-1.9
SIF_DestinationProvider	5.1.12-1.2
SIF_Element	5.1.11-1.4, 5.1.11-1.10, 5.1.12-1.6, 5.1.12-1.24, 5.1.12-1.29, 5.1.13-1.3
SIF_EncryptionLevel	5.1.3-1.7, 5.1.4-1.1, 5.3.3-1.78
SIF_Error	5.1.10-1.1, 5.2.1-1.6, 5.2.7-1.6, 5.2.21-1.10, 5.2.22-1.6, 5.2.23-1.8
SIF_Event	5.2.2-1.1
SIF_EventObject	5.2.2-1.4

SIF_Example	5.1.11–1.13
SIF_ExtendedDesc	5.1.10–1.5, 5.3.2–1.12
SIF_ExtendedElement	5.1.1–1.2
SIF_ExtendedElements	5.1.1–1.1, 5.3.1–1.51, 5.3.2–1.17, 5.3.3–1.122
SIF_ExtendedQuery	5.1.12–1.1, 5.2.6–1.6
SIF_ExtendedQueryResults	5.1.13–1.1, 5.2.7–1.8
SIF_ExtendedQuerySupport	5.2.3–1.5, 5.2.4–1.6, 5.2.4–1.27, 5.2.4–1.32, 5.3.3–1.15, 5.3.3–1.51, 5.3.3–1.59
SIF_From	5.1.12–1.9
SIF_GetAgentACL	5.2.15–1.1
SIF_GetMessage	5.2.13–1.1
SIF_GetZoneStatus	5.2.14–1.1
SIF_Header	5.1.3–1.1, 5.2.1–1.2, 5.2.2–1.2, 5.2.3–1.2, 5.2.4–1.2, 5.2.5–1.2, 5.2.6–1.2, 5.2.7–1.2, 5.2.8–1.2, 5.2.9–1.2, 5.2.18–1.2, 5.2.19–1.2, 5.2.20–1.2, 5.2.21–1.2, 5.2.22–1.2, 5.2.23–1.2, 5.3.2–1.5, 5.3.2–1.7
SIF_Icon	5.2.5–1.14, 5.3.3–1.4, 5.3.3–1.65
SIF_Join	5.1.12–1.11
SIF_JoinOn	5.1.12–1.13
SIF_LeftElement	5.1.12–1.14
SIF_LogEntry	5.3.2–1.1
SIF_LogEntryHeader	5.3.2–1.4
SIF_LogObject	5.3.2–1.14
SIF_LogObjects	5.3.2–1.13
SIF_MaxBufferSize	5.2.5–1.5, 5.2.6–1.4, 5.2.21–1.7, 5.3.3–1.79
SIF_Message	5.1.2–1.1
SIF_Metadata	5.3.1–1.50, 5.3.2–1.16, 5.3.3–1.121
SIF_Mode	5.2.5–1.6, 5.3.3–1.73
SIF_MorePackets	5.2.7–1.5, 5.2.21–1.9, 5.2.22–1.5, 5.2.23–1.7
SIF_MsgId	5.1.3–1.2
SIF_Name	5.1.8–1.6, 5.2.5–1.3, 5.3.3–1.3, 5.3.3–1.6, 5.3.3–1.64
SIF_NodeVendor	5.2.5–1.8, 5.3.3–1.66
SIF_NodeVersion	5.2.5–1.9, 5.3.3–1.67
SIF_NotificationType	5.2.16–1.2, 5.2.17–1.2
SIF_Object	5.2.3–1.3, 5.2.4–1.4, 5.2.4–1.9, 5.2.4–1.13, 5.2.4–1.17, 5.2.4–1.21, 5.2.4–1.25, 5.2.4–1.30, 5.2.8–1.3, 5.2.18–1.3, 5.2.20–1.3, 5.3.1–1.3, 5.3.1–1.7, 5.3.1–1.11, 5.3.1–1.15, 5.3.1–1.19, 5.3.1–1.23, 5.3.1–1.27, 5.3.3–1.13, 5.3.3–1.21, 5.3.3–1.28, 5.3.3–1.35, 5.3.3–1.42, 5.3.3–1.49, 5.3.3–1.57
SIF_ObjectData	5.2.2–1.3, 5.2.7–1.7
SIF_ObjectList	5.3.3–1.12, 5.3.3–1.20, 5.3.3–1.27, 5.3.3–1.34, 5.3.3–1.41, 5.3.3–1.48, 5.3.3–1.56
SIF_Operation	5.2.4–1.47, 5.2.4–1.53, 5.2.21–1.4, 5.2.23–1.4, 5.3.1–1.43, 5.3.1–1.49, 5.3.3–1.110, 5.3.3–1.119
SIF_Operations	5.2.4–1.46, 5.2.4–1.52, 5.3.1–1.42, 5.3.1–1.48, 5.3.3–1.109, 5.3.3–1.118
SIF_Operator	5.1.11–1.11, 5.1.12–1.26
SIF_OrderBy	5.1.12–1.28
SIF_OriginalHeader	5.3.2–1.6
SIF_OriginalMsgId	5.2.1–1.4
SIF_OriginalSourceId	5.2.1–1.3
SIF_PacketNumber	5.2.7–1.4, 5.2.21–1.8, 5.2.22–1.4, 5.2.23–1.6
SIF_Ping	5.2.10–1.1
SIF_Product	5.2.5–1.12, 5.3.3–1.7, 5.3.3–1.70
SIF_Property	5.1.8–1.5
SIF_Protocol	5.1.8–1.1, 5.2.5–1.7, 5.3.3–1.74, 5.3.3–1.84
SIF_ProtocolName	5.3.3–1.82
SIF_Provide	5.2.3–1.1
SIF_ProvideAccess	5.3.1–1.2
SIF_ProvideObjects	5.2.4–1.3
SIF_Provider	5.3.3–1.10
SIF_Providers	5.3.3–1.9

SIF_ProvideService	5.2.4–1.34, 5.3.1–1.30
SIF_Provision	5.2.4–1.1
SIF_PublishAddAccess	5.3.1–1.10
SIF_PublishAddObjects	5.2.4–1.12
SIF_PublishChangeAccess	5.3.1–1.14
SIF_PublishChangeObjects	5.2.4–1.16
SIF_PublishDeleteAccess	5.3.1–1.18
SIF_PublishDeleteObjects	5.2.4–1.20
SIF_Publisher	5.3.3–1.25, 5.3.3–1.32, 5.3.3–1.39
SIF_Query	5.1.11–1.1, 5.2.6–1.5
SIF_QueryObject	5.1.11–1.2
SIF_Register	5.2.5–1.1
SIF_Request	5.2.6–1.1
SIF_RequestAccess	5.3.1–1.22
SIF_Requester	5.3.3–1.54
SIF_Requesters	5.3.3–1.53
SIF_RequestMsgId	5.2.7–1.3, 5.2.16–1.4
SIF_RequestMsgIds	5.2.16–1.3
SIF_RequestObjects	5.2.4–1.24
SIF_RequestService	5.2.4–1.42, 5.3.1–1.38
SIF_RespondAccess	5.3.1–1.26
SIF_Responder	5.3.3–1.46
SIF_Responders	5.3.3–1.45
SIF_RespondObjects	5.2.4–1.29
SIF_RespondService	5.2.4–1.38, 5.3.1–1.34
SIF_Response	5.2.7–1.1
SIF_RightElement	5.1.12–1.16
SIF_Rows	5.1.13–1.7
SIF_SecureChannel	5.1.3–1.5
SIF_Security	5.1.3–1.4
SIF_Select	5.1.12–1.3
SIF_Service	5.2.4–1.35, 5.2.4–1.39, 5.2.4–1.43, 5.2.4–1.49, 5.2.21–1.3, 5.2.23–1.3, 5.3.1–1.31, 5.3.1–1.35, 5.3.1–1.39, 5.3.1–1.45, 5.3.3–1.93, 5.3.3–1.100, 5.3.3–1.107, 5.3.3–1.116
SIF_ServiceInput	5.2.21–1.1
SIF_ServiceList	5.3.3–1.92, 5.3.3–1.99, 5.3.3–1.106, 5.3.3–1.115
SIF_ServiceMsgId	5.2.17–1.4, 5.2.21–1.5, 5.2.22–1.3, 5.2.23–1.5
SIF_ServiceMsgIds	5.2.17–1.3
SIF_ServiceNotify	5.2.23–1.1
SIF_ServiceOutput	5.2.22–1.1
SIF_ServiceProvider	5.3.3–1.90
SIF_ServiceProviders	5.3.3–1.89
SIF_ServiceRequester	5.3.3–1.104
SIF_ServiceRequesters	5.3.3–1.103
SIF_ServiceResponder	5.3.3–1.97
SIF_ServiceResponders	5.3.3–1.96
SIF_ServiceSubscriber	5.3.3–1.113
SIF_ServiceSubscribers	5.3.3–1.112
SIF_SIFNode	5.3.3–1.62
SIF_SIFNodes	5.3.3–1.61
SIF_Sleep	5.2.11–1.1
SIF_Sleeping	5.3.3–1.80
SIF_SourceId	5.1.3–1.8, 5.3.3–1.72
SIF_Status	5.1.9–1.1, 5.2.1–1.5

SIF_Subscribe	5.2.8–1.1
SIF_SubscribeAccess	5.3.1–1.6
SIF_SubscribeObjects	5.2.4–1.8
SIF_Subscriber	5.3.3–1.18
SIF_Subscribers	5.3.3–1.17
SIF_SubscribeService	5.2.4–1.48, 5.3.1–1.44
SIF_SupportedAuthentication	5.3.3–1.81
SIF_SupportedProtocols	5.3.3–1.83
SIF_SupportedVersions	5.3.3–1.85
SIF_SystemControl	5.2.9–1.1
SIF_SystemControlData	5.2.9–1.3
SIF_Timestamp	5.1.3–1.3
SIF_Unprovide	5.2.18–1.1
SIF_Unregister	5.2.19–1.1
SIF_Unsubscribe	5.2.20–1.1
SIF_URL	5.1.8–1.4
SIF_Value	5.1.8–1.7, 5.1.11–1.12, 5.1.12–1.27
SIF_Vendor	5.2.5–1.11, 5.3.3–1.5, 5.3.3–1.69
SIF_Version	5.2.5–1.4, 5.2.5–1.13, 5.2.6–1.3, 5.2.21–1.6, 5.3.3–1.8, 5.3.3–1.71, 5.3.3–1.76, 5.3.3–1.86
SIF_VersionList	5.3.3–1.75
SIF_Wakeup	5.2.12–1.1
SIF_Where	5.1.12–1.18
SIF_ZoneStatus	5.3.3–1.1
TextData	A.1–1.4, A.2–1.5
URIOrBinaryType	A.15–1.1
URL	A.1–1.15, A.2–1.16
VersionType	A.16–1.1
VersionWithWildcardsType	A.17–1.1
XMLData	A.1–1.2, A.2–1.3

388 Total

Appendix M: Index of Attributes

Action	5.2.2-1.6
Alias	5.1.12-1.7, 5.1.13-1.5
Description	A.1-1.3, A.1-1.7, A.1-1.11, A.1-1.14, A.2-1.4, A.2-1.8, A.2-1.12, A.2-1.15
Distinct	5.1.12-1.4
FileName	A.1-1.6, A.1-1.10, A.2-1.7, A.2-1.11
LogLevel	5.3.2-1.3
MIMEType	A.1-1.5, A.1-1.9, A.1-1.13, A.2-1.6, A.2-1.10, A.2-1.14
Name	5.1.1-1.3
ObjectName	5.1.11-1.3, 5.1.12-1.8, 5.1.12-1.10, 5.1.12-1.15, 5.1.12-1.17, 5.1.12-1.25, 5.1.12-1.30, 5.1.13-1.4, 5.2.2-1.5, 5.2.3-1.4, 5.2.4-1.5, 5.2.4-1.10, 5.2.4-1.14, 5.2.4-1.18, 5.2.4-1.22, 5.2.4-1.26, 5.2.4-1.31, 5.2.8-1.4, 5.2.18-1.4, 5.2.20-1.4, 5.3.1-1.4, 5.3.1-1.8, 5.3.1-1.12, 5.3.1-1.16, 5.3.1-1.20, 5.3.1-1.24, 5.3.1-1.28, 5.3.2-1.15, 5.3.3-1.14, 5.3.3-1.22, 5.3.3-1.29, 5.3.3-1.36, 5.3.3-1.43, 5.3.3-1.50, 5.3.3-1.58
Ordering	5.1.12-1.31
RefId	A.2-1.2
RowCount	5.1.12-1.5
Secure	5.1.8-1.3
ServiceName	5.2.4-1.36, 5.2.4-1.40, 5.2.4-1.44, 5.2.4-1.50, 5.3.1-1.32, 5.3.1-1.36, 5.3.1-1.40, 5.3.1-1.46, 5.3.3-1.94, 5.3.3-1.101, 5.3.3-1.108, 5.3.3-1.117
SIF_Action	5.1.1-1.5
Source	5.3.2-1.2
SourceId	5.3.3-1.11, 5.3.3-1.19, 5.3.3-1.26, 5.3.3-1.33, 5.3.3-1.40, 5.3.3-1.47, 5.3.3-1.55, 5.3.3-1.91, 5.3.3-1.98, 5.3.3-1.105, 5.3.3-1.114
Type	5.1.8-1.2, 5.1.11-1.6, 5.1.11-1.8, 5.1.12-1.12, 5.1.12-1.20, 5.1.12-1.22, 5.3.3-1.63
Version	5.1.2-1.3
xmlns	5.1.2-1.2
xsi:type	5.1.1-1.4, 5.1.13-1.6
ZoneId	5.3.3-1.2

100 Total

Appendix N: References

Architecture/Infrastructure

Key	Citation
EXPORT	U.S. Bureau of Industry and Security. Commercial Encryption Export Controls. 6 July 2006 < http://www.bis.doc.gov/Encryption/ >.
FAVICON	icon - Wikipedia, the free encyclopedia. 6 July 2006 < http://en.wikipedia.org/wiki/Favicon >.
MIME	IETF (Internet Engineering Task Force). RFC 2048: Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures. 1996 November. 6 July 2006 < http://www.ietf.org/rfc/rfc2048.txt >.
RFC 2045	IETF (Internet Engineering Task Force). RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. 6 July 2006 < http://www.ietf.org/rfc/rfc2045.txt >.
RFC 2046	IETF (Internet Engineering Task Force). RFC 2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. 6 July 2006 < http://www.ietf.org/rfc/rfc2046.txt >.
RFC 2119	IETF (Internet Engineering Task Force). RFC 2119: Key words for use in RFCs to Indicate Requirement Levels. 11 December 2008 < http://www.ietf.org/rfc/rfc2119.txt >.
RFC 2246	IETF (Internet Engineering Task Force). RFC 2246: The TLS Protocol: Version 1.0. 6 July 2006 < http://www.ietf.org/rfc/rfc2246.txt >.
RFC 2376	IETF (Internet Engineering Task Force): RFC 2376: XML Media Types. 6 July 2006 < http://www.ietf.org/rfc/rfc2376.txt >.
RFC 2396	IETF (Internet Engineering Task Force): RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax. 13 July 2006 < http://www.ietf.org/rfc/rfc2396.txt >.
RFC 2518	IETF (Internet Engineering Task Force). RFC 2518: HTTP Extensions for Distributed Authority—WEBDAV. 6 July 2006 < http://www.ietf.org/rfc/rfc2518.txt >.
RFC 2616	IETF (Internet Engineering Task Force). RFC 2616: Hypertext Transport Protocol—HTTP 1.1. 6 July 2006 < http://www.ietf.org/rfc/rfc2616.txt >.
RFC 4122	IETF (Internet Engineering Task Force). A Universally Unique Identifier (UUID) URN Namespace. 3 July 2006 < http://www.ietf.org/rfc/rfc4122.txt >.

Key	Citation
SCHEMA	<p>World Wide Web Consortium (W3C). XML Schema Part 1: Structures. 6 July 2006 <http://www.w3.org/TR/xmlschema-1/>.</p> <p>World Wide Web Consortium (W3C). XML Schema Part 2: Datatypes. 6 July 2006 <http://www.w3.org/TR/xmlschema-2/>.</p> <p>A non-normative primer on XML Schema is also available:</p> <p>World Wide Web Consortium (W3C). XML Schema Part 0: Primer. 6 July 2006 <http://www.w3.org/TR/xmlschema-0/>.</p>
Schneier	Schneier, Bruce. Applied Cryptography: Protocols, Algorithms, and Source Code in C (Second Edition). John Wiley & Sons, 1995.
SIF Certification	Schools Interoperability Framework Association (SIF Association). SIF Certification - Product Standards. 23 May 2007 < http://certification.sifinfo.org/docs/prodstandards.tpl >.
SIF Reporting WS	Schools Interoperability Framework Association (SIF Association). Schools Interoperability Framework™ Reporting Web Service 1.0. 28 September 2006 < http://specification.sifinfo.org/WebServices/Reporting/1.0 >.
SSL2	Netscape. SSL 2.0 Protocol Specification. 6 July 2006 < http://wp.netscape.com/eng/security/SSL_2.html >.
SSL3	Netscape. The SSL Protocol Version 3.0. 6 July 2006 < http://wp.netscape.com/eng/ssl3/draft302.txt >.
WSARCH	World Wide Web Consortium (W3C). Web Services Architecture. 16 July 2006 < http://www.w3.org/TR/ws-arch/ >.
XML	W3C (World Wide Web Consortium). Extensible Markup Language (XML) 1.0 (Third Edition). 6 July 2006 < http://www.w3.org/TR/2004/REC-xml-20040204 >.
XMLINTRO	W3C (World Wide Web Consortium). XML in 10 Points. 13 July 2006 < http://www.w3.org/XML/1999/XML-in-10-points.html >.
XMLNS	W3C (World Wide Web Consortium). Namespaces in XML. 6 July 2006 < http://www.w3.org/TR/REC-xml-names/ >.
XPATH	W3C (World Wide Web Consortium). XML Path Language (XPath) Version 1.0. 6 July 2006 < http://www.w3.org/TR/xpath >.