Contents

[**2 Introduction** 3](#_Toc271175858)

[**2.1 Specification Organization** 4](#_Toc271175859)

[**2.2 Document Conventions** 4](#_Toc271175860)

[**2.2.1 Definitions** 4](#_Toc271175861)

[**2.2.2 Structure and Values** 4](#_Toc271175862)

[**2.2.3 Examples** 5](#_Toc271175863)

[**2.2.4 References** 5](#_Toc271175864)

[**2.2.5 Terminology** 5](#_Toc271175865)

[**2.2.6 XML Diagrams** 5](#_Toc271175866)

[**2.3 Version Numbers** 7](#_Toc271175867)

[**2.4 Highlighted Additions/Changes Since Version 2.3** 8](#_Toc271175868)

[**2.4.1 Zone Services** 8](#_Toc271175869)

[**2.4.2 Data Model** 9](#_Toc271175870)

[**2.4.3 Content Catalog** 11](#_Toc271175871)

[**2.4.4 Enhanced Security** 12](#_Toc271175872)

[**2.4.5 XSD Updates** 12](#_Toc271175873)

[3 Architecture 12](#_Toc271175874)

[3.1 Assumptions 12](#_Toc271175875)

[3.1.1 Notes on Related Technologies 13](#_Toc271175876)

[3.2 Concepts 13](#_Toc271175877)

[3.2.1 Data Model 14](#_Toc271175878)

[3.2.2 Zone Architecture 14](#_Toc271175879)

[3.2.3 Infrastructure and Messaging 15](#_Toc271175880)

[3.2.4 Data Provision: A Request/Response Model 16](#_Toc271175881)

[3.2.5 Event Reporting: A Publish/Subscribe Model 17](#_Toc271175882)

[3.2.6 Communication: An Asynchronous Model 19](#_Toc271175883)

[3.2.7 Security Model 20](#_Toc271175884)

[3.2.8 Zone Services 22](#_Toc271175885)

[3.3 SIF Architecture 25](#_Toc271175886)

[3.3.1 Architectural Components 25](#_Toc271175887)

[3.3.2 Communications Protocols 29](#_Toc271175888)

[3.3.2.1 Naming Conventions for Agents and Zone Integration Servers 30](#_Toc271175889)

[3.3.2.2 Object Identifiers 30](#_Toc271175890)

[3.3.4 Agent/Application Requirements 31](#_Toc271175891)

[3.3.5 Zone Integration Server Requirements 37](#_Toc271175892)

[3.3.6 Message Processing 47](#_Toc271175893)

[3.3.7 Infrastructure Transport Layer 59](#_Toc271175894)

[4 Messaging 68](#_Toc271175895)

[4.1 Agent Protocols 68](#_Toc271175896)

[4.1.1 Agent Messaging Protocols 69](#_Toc271175897)

[4.1.2 Agent Message Handling Protocols 116](#_Toc271175898)

[4.2 ZIS Protocols 152](#_Toc271175899)

[4.2.1 ZIS Messaging Protocols 152](#_Toc271175900)

[4.2.2 ZIS Message Handling Protocols 160](#_Toc271175901)

[5 Infrastructure 199](#_Toc271175902)

[5.1 Common Elements 199](#_Toc271175903)

[5.1.1 SIF\_Message 200](#_Toc271175904)

[5.1.2 SIF\_Header 203](#_Toc271175905)

[5.1.3 SIF\_EncryptionLevel 208](#_Toc271175906)

[5.1.4 SIF\_AuthenticationLevel 209](#_Toc271175907)

[5.1.5 SIF\_Contexts 210](#_Toc271175908)

[5.1.6 SIF\_Context 210](#_Toc271175909)

[5.1.7 SIF\_Protocol 211](#_Toc271175910)

[5.1.8 SIF\_Status 212](#_Toc271175911)

[5.1.9 SIF\_Error 213](#_Toc271175912)

[5.1.10 SIF\_Query 216](#_Toc271175913)

[5.1.11 SIF\_ExtendedQuery 224](#_Toc271175914)

[5.1.12 SIF\_ExtendedQueryResults 238](#_Toc271175915)

[5.2 Messages 241](#_Toc271175916)

[5.2.1 SIF\_Ack 241](#_Toc271175917)

[5.2.2 SIF\_Event 243](#_Toc271175918)

[5.2.3 SIF\_Provide 245](#_Toc271175919)

[5.2.4 SIF\_Provision 247](#_Toc271175920)

[5.2.5 SIF\_Register 258](#_Toc271175921)

[5.2.6 SIF\_Request 263](#_Toc271175922)

[5.2.7 SIF\_Response 265](#_Toc271175923)

[5.2.8 SIF\_Subscribe 271](#_Toc271175924)

[5.2.9 SIF\_SystemControl 272](#_Toc271175925)

[5.2.10 SIF\_Ping 274](#_Toc271175926)

[5.2.11 SIF\_Sleep 276](#_Toc271175927)

[5.2.12 SIF\_Wakeup 277](#_Toc271175928)

[5.2.13 SIF\_GetMessage 279](#_Toc271175929)

[5.2.14 SIF\_GetZoneStatus 281](#_Toc271175930)

[5.2.15 SIF\_GetAgentACL 282](#_Toc271175931)

[5.2.16 SIF\_CancelRequests 282](#_Toc271175932)

[5.2.17 SIF\_CancelServiceInputs 284](#_Toc271175933)

[5.2.18 SIF\_Unprovide 285](#_Toc271175934)

[5.2.19 SIF\_Unregister 287](#_Toc271175935)

[5.2.20 SIF\_Unsubscribe 287](#_Toc271175936)

[5.2.21 SIF\_ServiceInput 289](#_Toc271175937)

[5.2.22 SIF\_ServiceOutput 297](#_Toc271175938)

[5.2.23 SIF\_ServiceNotify 301](#_Toc271175939)

[5.3 Objects 307](#_Toc271175940)

[5.3.1 SIF\_AgentACL 307](#_Toc271175941)

[5.3.2 SIF\_LogEntry 316](#_Toc271175942)

[5.3.3 SIF\_ZoneStatus 325](#_Toc271175943)

[7 Zone Services 349](#_Toc271175944)

[7.1 Introduction 349](#_Toc271175945)

[7.1.1 Important Terms 349](#_Toc271175946)

[7.1.2 New Message Types 351](#_Toc271175947)

[7.1.3 SIF Design Requirements for Zone Service Technology 353](#_Toc271175948)

[7.1.4 Zone Service Design Best Practices 353](#_Toc271175949)

[7.1.5 Zone Service Usage Best Practices 354](#_Toc271175950)

**2 Introduction**

**2.1 Specification Organization**

Beyond the abstract and this introduction, educators and non-technical readers are typically interested in the pK-12 data objects that can be shared and reported on by SIF-enabled applications in SIF implementations. These are presented in the [Data Model](http://specification.sifinfo.org/Implementation/2.4/DataModel.html) section in a format that should not be a barrier to readers with a background that includes a brief introduction to [[XML]](http://specification.sifinfo.org/Implementation/2.4/References.html#XML), though they may benefit from the introductory sections of [Architecture](http://specification.sifinfo.org/Implementation/2.4/Architecture.html). Technical readers, including software architects, developers and integrators, should have a solid background in [Architecture](http://specification.sifinfo.org/Implementation/2.4/Architecture.html), [Messaging](http://specification.sifinfo.org/Implementation/2.4/Messaging.html), [Infrastructure](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html) and [Data Model](http://specification.sifinfo.org/Implementation/2.4/DataModel.html).

* The [Preamble](http://specification.sifinfo.org/Implementation/2.4/index.html#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](http://specification.sifinfo.org/Implementation/2.4/Architecture.html) describes the assumptions, concepts, models, and requirements related to the SIF infrastructure and data model.
* [Messaging](http://specification.sifinfo.org/Implementation/2.4/Messaging.html) details the actions Agents and Zone Integration Servers take when sending and receiving messages.
* [Infrastructure](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html) 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 [Data Model](http://specification.sifinfo.org/Implementation/2.4/DataModel.html) section provides definitions of the XML structure for common elements in the data model and all objects related to entities in the pK-12 environment. This section is organized by the working groups and task forces within the SIF Association that have defined common elements or objects.
* [Zone Services](http://specification.sifinfo.org/Implementation/2.4/ZoneServices.html) details the data formats and operation choreographies of the interfaces which Zone Services provide in support of their clients.
* 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](http://specification.sifinfo.org/Implementation/2.4/References.html) 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**

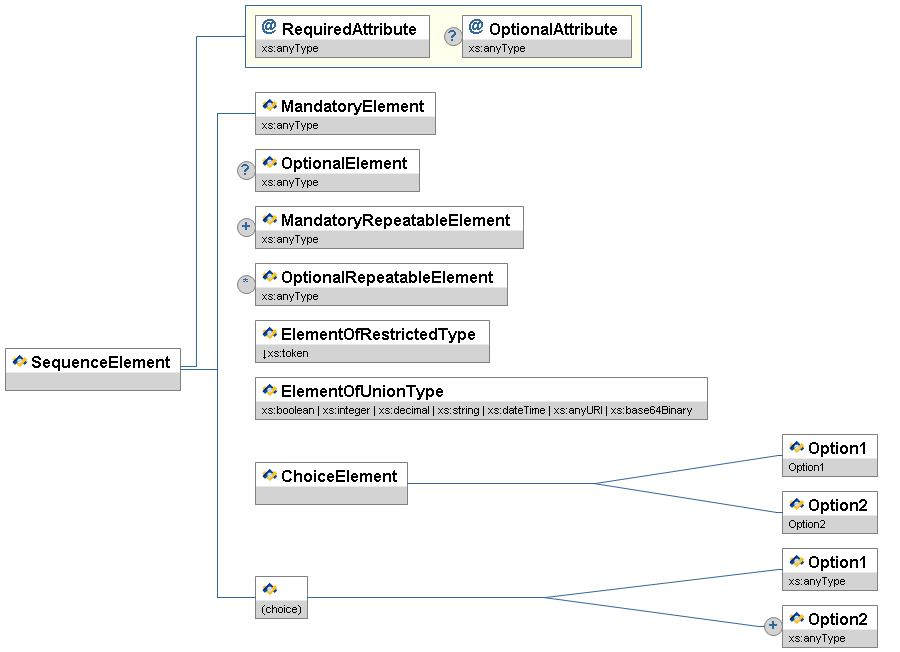
References to other works occurring in this text are given in brackets, e.g. [REFERENCE]. The text in brackets corresponds to a key in the [References](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "References) appendix. Often when the text in the brackets duplicates surrounding 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2119).

**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* r *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 2.3**

This release contains the following significant updates and extensions to the SIF specification.

**2.4.1 Zone Services**

Zone Services is the third major transactional model of the SIF Implementation Specification; joining the publish/subscribe and the request/response functionality. Zone Services provide its clients (existing SIF applications and other Zone Services) with the following capabilities:

* **Customized Interfaces**

The supported operations and notifications of a Zone Service can extend beyond the “create/read/update/delete” (CRUD) data interface of an object provider.

* **Process Choreographies**

A Zone Service interface can encapsulate the implementation details of a stateful process that includes multiple transactions.

* **Customized Data Collections**

A Zone Service response can package together data elements from multiple data objects of different types, effectively encapsulating the SIF Data Object hierarchy.

Four new messages were added to the SIF infrastructure to support these capabilities: ServiceInput, ServiceOutput, ServiceNotification, and CancelServiceInputs. Message processing choreographies are described in Section 4: Messaging and message contents are defined in Section 5: Infrastructure.

Three sets of Zone Services are included with this release, all of which are defined in Section 7: Zone Services.

* Student and Staff Identifier
* Assessment
* Student Record Exchange

The specific changes are detailed below.

* Added SIF Zone Services Infrastructure
* Added new SIF Zone Service serviceAgencyStudentIdRequest
* Added new SIF Zone Service serviceAgencyStudentIdManagement
* Added new SIF Zone Service serviceAgencyStaffIdRequest
* Added new SIF Zone Service serviceAgencyStaffIdManagement
* Added new SIF Zone Service serviceAssessmentProcessing
* Added new SIF Zone Service serviceAssessmentAdministration
* Added new SIF Zone Service serviceSREBroker
* Added new SIF Zone Service serviceSREPublisher
* Added new SIF Zone Service serviceSREConsumer
* Added new SIF Zone Service serviceSREDemographicRecordConsumer
* Added new SIF Zone Service serviceSREAcademicRecordConsumer
* Added new SIF Zone Service serviceSRESpecialEducationRecordConsumer

**2.4.2 Data Model**

Data Model changes were made for Instructional Services, Student Record Exchange, Assessment, Special Programs, Student Information Systems and Metadata. The specific changes are detailed below.

* Added code set value(s) to 1048 Secondary Incident Behavior
* Added code set value(s) to LearningStandardDocument/Source
* Added code set value(s) to ProgramStatus/Code
* Added code set value(s) to StudentPlacement/FrequencyTime/Code
* Added common element ExtendedTechnicalRequirements
* Added common element OtherIdList
* Added common element PersonInfo
* Added element Activity/ExtendedTechnicalRequirements
* Added element AssessmentItem/ItemScoreMaximum
* Added element AssessmentItem/ItemScoreMinimum
* Added element AssessmentItem/PerformanceLevels
* Added element AssessmentSubTest/Abbreviation
* Added element AssessmentSubTest/ContainerOnly
* Added element AssessmentSubTest/Description
* Added element AssessmentSubTest/NumberOfItems
* Added element Assignment/ExtendedTechnicalRequirements
* Added element CalendarDate/BellScheduleDayList
* Added element Demographics/FirstUSEnrollment
* Added element DisciplineIncident/IncidentActionList
* Added element DisciplineIncident/IncidentDescription
* Added element DisciplineIncident/OffenderList/Offender/ManifestationDetermination/MeetingOutcome
* Added element DisciplineIncident/OffenderList/Offender/RelatedToList
* Added element DisciplineIncident/OffenderList/Offender/WeaponTypeList
* Added element EducationFilter/LearningObjectives
* Added element EducationFilter/ResourceTypes
* Added element EducationFilter/Summary
* Added element EducationFilter/Title
* Added element LearningStandardItem/Level4
* Added element LearningStandardItem/Level5
* Added element MeetingTime/TimetableDay
* Added element MeetingTime/TimetablePeriod
* Added element SectionInfo/BellScheduleInfoRefId
* Added element SectionInfo/CountForAttendance
* Added element SectionInfo/CourseSection
* Added element SectionInfo/SectionCode
* Added element StudentAcademicRecord/CourseHistory/Term/Courses/Course/MarkingPeriods/MarkingPeriod/CourseCreditsEarned
* Added element StudentAcademicRecord/CourseHistory/Term/Courses/Course/SectionInfoData
* Added element StudentAcademicRecord/EnrollmentHistory/StudentSchoolEnrollmentData/EntryType
* Added element StudentAcademicRecord/EnrollmentHistory/StudentSchoolEnrollmentData/ExitStatus
* Added element StudentAcademicRecord/EnrollmentHistory/StudentSchoolEnrollmentData/ExitType
* Added element StudentAcademicRecord/EnrollmentHistory/StudentSchoolEnrollmentData/GradeLevel
* Added element StudentAcademicRecord/EnrollmentHistory/StudentSchoolEnrollmentData/StudentLEARelationship
* Added element StudentAcademicRecord/EnrollmentHistory/StudentSchoolEnrollmentData/StudentSchoolRelationship
* Added element StudentAttendanceSummary/StudentSchoolEnrollmentRefId
* Added element StudentDemographicRecord/StudentPersonalData/Demographics/CountryOfBirth
* Added element StudentDemographicRecord/StudentPersonalData/Demographics/DwellingArrangement
* Added element StudentDemographicRecord/StudentPersonalData/Demographics/HispanicLatino
* Added element StudentParticipation/AdditionalProgramAvailabilityList
* Added element StudentParticipation/CaseManager
* Added element StudentResponseSet/Items/Item/ItemAidSetUsed
* Added element StudentResponseSet/Items/Item/ItemScore
* Added element StudentSchoolEnrollment/PromotionInfo/Nonpromotion
* Added element StudentSchoolEnrollment/PromotionInfo/Promotion
* Added element StudentScoreSet/Scores/Score/Description
* Added element StudentScoreSet/Scores/Score/NumberOfResponses
* Added element StudentSpecialEducationRecord/StudentParticipationData/StudentSpecialEducationFTE
* Added element StudentSpecialEducationRecord/StudentParticipationList
* Added element StudentSpecialEducationRecord/StudentPlacementList
* Added object BellSchedule
* Added object ContentCatalog
* Added object StudentContactPersonal
* Added object StudentContactRelationship
* Added object StudentSpecialEducationSummary
* Corrected description EmployeeContract/BaseSalary
* Updated element description StudentAttendanceSummary/DaysAttended
* Updated element description StudentAttendanceSummary/DaysInMembership
* Updated element description StudentAttendanceSummary/EndDate
* Updated element description StudentAttendanceSummary/EndDay
* Updated element description StudentAttendanceSummary/ExcusedAbsences
* Updated element description StudentAttendanceSummary/FTE
* Updated element description StudentAttendanceSummary/Resident
* Updated element description StudentAttendanceSummary/StartDate
* Updated element description StudentAttendanceSummary/StartDay
* Updated element description StudentAttendanceSummary/UnexcusedAbsences
* Updated element description StudentAttendanceSummary
* Updated element type StudentContactPersonal/EducationalLevel

**2.4.3 Content Catalog**

The addition of the ContentCatalog object makes it possible to search for relevant instructional assets that reside in multiple content repositories, and return one catalog of those resources.

**2.4.4 Enhanced Security**

An XML Filtering capability on both elements and messages has been defined for the ZIS to allow it to implement site-specific data security policies.

**2.4.5 XSD Updates**

The way that a choice block is implemented in Data Model objects has been changed. Since SIF Implementation Specification 2.0r1, a choice of elements has been implemented in the schema files as a sequence of optional elements. The requirement to include just one, and only one, of the elements of the choice block has not been enforced in the schema. Infrastructure objects and messages do not use this approach.

Beginning with version 2.4, the Data Model schema will fully enforce choice blocks by utilizing the xs:choice tag. Agents implementing the Specification as written will not be affected. The following Data Model object schemas will change slightly:

* FoodserviceTransaction
* ReportManifest
* SIF\_ReportObject
* AbstractContentElementType
* AbstractContentPackageType

# 3 Architecture

## 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "XML) and its use. Readers lacking this prerequisite are referred to [[XMLINTRO]](http://specification.sifinfo.org/Implementation/2.4/References.html#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]](http://specification.sifinfo.org/Implementation/2.4/References.html#RFC2616) referenced in the SIF HTTPS and SIF HTTP Transport Layers.
* Those portions of TLS 1.0 [[RFC 2246]](http://specification.sifinfo.org/Implementation/2.4/References.html#RFC2246), SSL 3.0 [[SSL3]](http://specification.sifinfo.org/Implementation/2.4/References.html#SSL3) and SSL 2.0 [[SSL2]](http://specification.sifinfo.org/Implementation/2.4/References.html#SSL3) 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]](http://specification.sifinfo.org/Implementation/2.4/References.html#XML) and its references to Unicode and the UTF-8 Encoding.
* The role and use of namespaces in XML [[XMLNS]](http://specification.sifinfo.org/Implementation/2.4/References.html#XMLNS).
* Accessing XML elements/attributes using XPath [[XPATH]](http://specification.sifinfo.org/Implementation/2.4/References.html#XPATH).
* XML Schema data types and structures [[XMLSCHEMA]](http://specification.sifinfo.org/Implementation/2.4/References.html#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](http://specification.sifinfo.org/Implementation/2.4/NotesOnRelatedTechnologies.html), 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 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.



Figure : Zone Architecture Block Diagram

### 3.2.1 Data Model

Data exchanged in SIF is defined using a series of **data objects**. These objects are expressed using [[XML]](http://specification.sifinfo.org/Implementation/2.4/References.html#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 asStudentPersonal , StudentSchoolEnrollment and StaffPersonal

### 3.2.2 Zone Architecture

The common feature of all SIF topographies is that a number of applications wish to share data. 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. Agents then communicate with other agents by routing these data objects througha ZIS. One ZIS can communicate with other zone by sending data object to the other zone’s ZIS. Through this mechanism one can create a complex infrastructure for sharing information across schools, districts, states, etc.

To avoid extensive and potentially erroneous access to information, the ZIS also provides access control so the Zone Adminstrator can control which applications have access to which SIF data.

#### 3.2.2.1 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 customers and system integrators to work with data in new ways while retaining the zone topologies.

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 included in this 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.

### 3.2.3 Infrastructure and Messaging



Figure 2: Pub/Sub & Request/Response Messaging Paradigms

Agents share data in a Zone via two models, the Publish/Subscribe model and the Request/Response model. Agents publish data changes of interest to subscribers by sending a SIF\_Event message to the ZIS. Agents can also request or query data from other agents in a Zone by sending a SIF\_Request message to an Agent, eventually being sent one or more SIF\_Response messages in return. This exchange of messages over a SIF-defined transport layer, SIF HTTPS or SIF HTTP, is the primary feature that defines the SIF Infrastructure.



Figure 3: SIF\_Message Structure

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.

### 3.2.4 Data Provision: A Request/Response Model

**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.

**Access Control System**

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.

**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.

### 3.2.5 Event Reporting: A Publish/Subscribe Model



Figure 4: Subscribing for Events

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.

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 5 ZIS Registration

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.



Figure 6: Multiple Publishers for a Data Object

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.

**Business Rules?**

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.

### 3.2.6 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.

**Guaranteed Delivery**

By requiring asynchronous communications, a SIF implementation can exploit software designs that achieve high scalability and reliability. For example, even if an agent is not currently connected to a Zone, another application's agent can still send messages to that agent knowing that the ZIS will deliver those messages as soon as the agent is available.



Figure 7: Synchronous Communication to the ZIS

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.

### 3.2.7 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 8: SIF HTTPS Communication Protocol

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.

#### 3.2.7.1 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.

#### 3.2.7.2 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.

#### 3.2.7.3 Access Control



Figure 9: 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.

### 3.2.8 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.



Figure 10: Conceptual Zone Service

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 3.2.8-1: Differences between a Data Object and a Zone Service

## 3.3 SIF 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](http://specification.sifinfo.org/Implementation/2.4/Messaging.html) and [Infrastructure](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html).

### 3.3.1 Architectural Components



Figure 11: SIF Zone

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 architecture. It 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.



Figure 12: SIF Zone Example

Each application requires an agent, who 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.

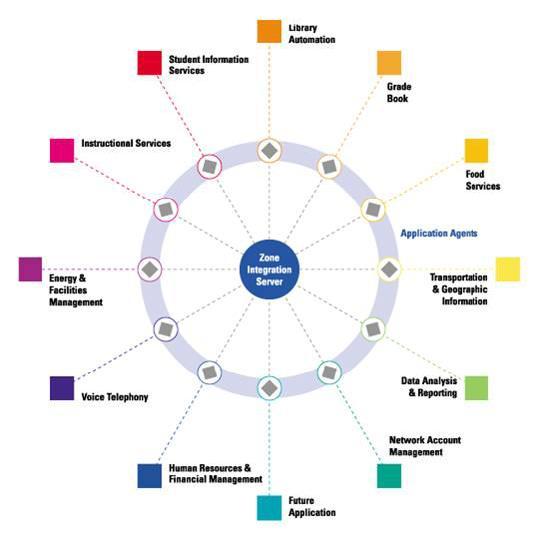


Figure 3.3.1-1: Single-Zone School SIF Implementation

**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.

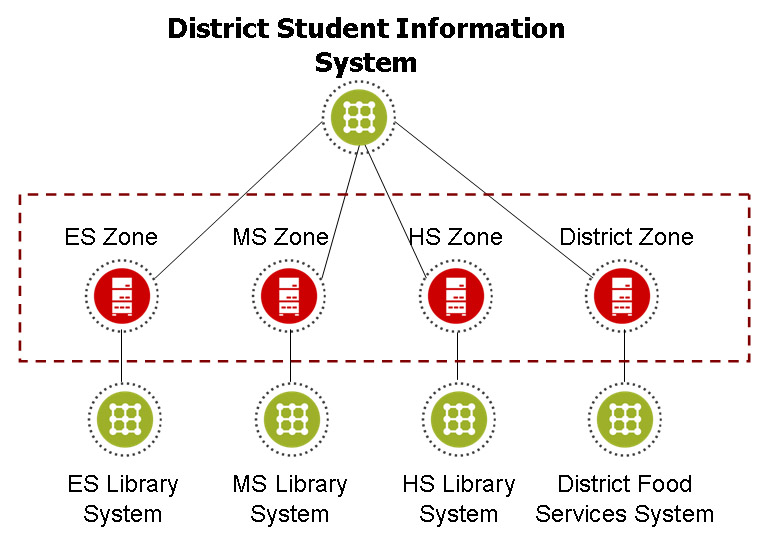


Figure 3.3.1-2: Multiple-Zone District SIF Implementation

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 date up to the state, or an application that supports StudentLocator, managing state-level student identifiers, and so on.

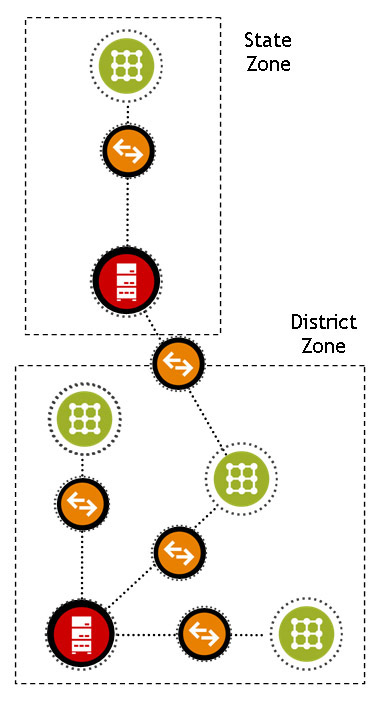


Figure 3.3.1-3: Multiple-Zone State SIF Implementation

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.

### 3.3.2 Communications Protocols

### 3.3.2.1 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.

### 3.3.2.2 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 UUIDs 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 RefIds 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 (UUID) that MUST be generated per published algorithms [[RFC 4122]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC4122) whenever a RefId is used. UUIDs 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 UUID 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 RefIds can be found in [Data Model](http://specification.sifinfo.org/Implementation/2.4/DataModel.html).

#### 3.3.2.2.1 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).

### Message Structure3.3.4 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](http://specification.sifinfo.org/Implementation/2.4/Messaging.html) and [Infrastructure](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html).

#### 3.3.4.1 Communicate with the ZIS

SIF HTTPS

Support for [SIF HTTPS](http://specification.sifinfo.org/Implementation/2.4/Architecture.html" \l "SIFHTTPSTransport) 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](http://specification.sifinfo.org/Implementation/2.4/Architecture.html#SIFHTTPTransport) 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.

ZIF – 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.

#### 3.3.4.2 Transmit Application Changes to the ZIS



Figure 13: Transmit Application Changes

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.

If an application does not support an optional field of an object or that element was not changed during the last edit, it MUST NOT send an empty element, (eg., <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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html) for the object.

#### 3.3.4.3 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.

<SIF\_Response ...>  
 <SIF\_Header>  
... </SIF\_Header>  
…

<SIF\_PacketNumber>1</SIF\_PacketNumber>  
 <SIF\_MorePackets>No</SIF\_MorePackets>  
 <SIF\_Error>  
 <SIF\_Category>??</SIF\_Category>  
 <SIF\_Code>??</SIF\_Code>  
 <SIF\_Desc>invalid object</SIF\_Desc>  
 </SIF\_Error>  
</SIF\_Response>

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.

#### 3.3.4.4 Vendor's 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 UUID, before releasing that object to the zone in an Add event or in a SIF\_Response. For example, 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, (i.e. <Name Type="04 "/> or <Name Type="04 "></Name> ); the fields MUST be omitted from the XML stream instead.

#### 3.3.4.5 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.

#### 3.3.4.6 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.

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

#### 3.3.4.7 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 has at least one version in common with all registered Agents they can then communicate. It is 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](http://specification.sifinfo.org/Implementation/2.4/Architecture.html" \l "MultipleVersionZones).

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Register__SIF_Version) 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](http://specification.sifinfo.org/Implementation/2.4/WildcardVersionSupportImplementationNotes.html).

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]](http://specification.sifinfo.org/Implementation/2.4/References.html#SIFCertification) associated with a major release cycle, if application vendors wish to establish their applications as SIF Certified™.

### 3.3.5 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, this example describes a database consisting of an Access Control List and Zone Status.

#### 3.3.5.1 Access Control List (ACL)



Figure 14: Access Control List Conceptual Diagram

First 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 SIF\_Register message) |
| Register | May this agent register in the zone? |

Table 3.3.5.1-1: 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 3.3.5.1-2: 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 SIF\_Register message) |
| Context Name | The name of the SIF Context to which the permissions apply |
| Object Name | The object being manipulated (e.g., StudentPersonal , 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 3.3.5.1-3: 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 3.3.5.1-4: 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.

#### 3.3.5.2 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 optional 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "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 | //StudentPersonal/MedicalAlertMessages |
| AcmeLibrary | //StudentPersonal/IDEA |
| AcmeSIS | SIF\_Message[SIF\_Event/SIF\_Header/SIF\_SourceId="AcmeLibrary"]//StudentPersonal/LocalId |

Table 3.3.5.2-1: 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 | SIF\_Message[/SIF\_Message/SIF\_Event/SIF\_ObjectData/SIF\_EventObject/SIF\_LogEntry[@Source="Agent"]/SIF\_OriginalHeader/SIF\_Header[SIF\_SourceId!="AcmeLibrary"]] |
| AcmeTrans | SIF\_Message[/SIF\_Message/SIF\_Event/SIF\_ObjectData/SIF\_EventObject/SIF\_LogEntry[@Source="Agent"]/SIF\_OriginalHeader/SIF\_Header[SIF\_SourceId!="AcmeTrans"]] |
| AcmeSIS | SIF\_Message[/SIF\_Message/SIF\_Event/SIF\_ObjectData/SIF\_EventObject/SIF\_LogEntry[@Source="Agent"]/SIF\_OriginalHeader/SIF\_Header[SIF\_SourceId!="AcmeSIS"]] |

Table 3.3.5.2-2: XML Filter Example SIF\_LogEntry

##### 3.3.5.2.1 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 UUIDelines.

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.

##### 3.3.5.2.2 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.

#### 3.3.5.3 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ZoneStatus) for requirements.

#### 3.3.5.4 Zone Context Registry



Figure 15: Contexts Relationships Example

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 2.0 each zone will, at a minimum, support the SIF\_Default context. A ZIS MAY support allowing additional contexts to be defined within the context registry at the discretion of the ZIS administrator.

In addition to setting up contexts, Zone Administrators can define agent access to each object/context through the Access Control List. . The set of permissions for each agent within each context MUST be able to be set differently than ACL permissions within a different context for the same agent.

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.

#### 3.3.5.5 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.

#### 3.3.5.6 Support Selective Message Blocking (SMB) to Resolve Deadlocks

##### 3.3.5.6.1 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 by the ZIS to request information from other Agents while processing a SIF\_Event message, without causing communication "deadlock" between an agent and a ZIS.



Figure 16: SMB Event Message Timing

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.

##### 3.3.5.6.2 Requirements



Figure 17: SMB choreorgraphy

* 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.

##### 3.3.5.6.3 Example

For a detailed example of SMB, see the [Selective Message Blocking (SMB) Example](http://specification.sifinfo.org/Implementation/2.4/SelectiveMessageBlockingSMBExample.html).

#### 3.3.5.7 Quality of Service Implementation



Figure 18: Buffered Messages

When a message is large, it can be broken into packets for transmission. For quality of service the ZIS makes sure all the messages are delivered to the registered agents. This requires the ZIS to buffer all the packets to make sure it receives them all and then to further transmit them to the agents.

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. Once the message stream has been completed, if the agent initiating the buffered message stream attempts to send any more messages, it 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 a configurable amount of time.If a ZIS does remove tracking information for a message, it MUST notify any registered agents of the removal.

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.

### 3.3.6 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.

#### 3.3.6.1 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "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 which 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 "doctypedecl" (<!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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html) and [Data Model](http://specification.sifinfo.org/Implementation/2.4/DataModel.html). 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 and they choose to turn that feature on. 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.

#### 3.3.6.2 Message Identification

Each message originating from an agent or ZIS needs to 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 universally unique identifier [[RFC 4122]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC4122) 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 UUUIDs, see [[RFC 4122]](http://specification.sifinfo.org/Implementation/2.4/References.html#RFC4122).

#### 3.3.6.3 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.

##### 3.3.6.3.1 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.

##### 3.3.6.3.2 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. A Zone administrator can configure the ZIS to a higher minimum encryption and authentication levels for the Zone.

The receiving agent must have registered to receive either the minimum level set by the administrator and/or the level published in the message. 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 insecure messages. The ZIS administrator needs to configure the zone 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.

##### 3.3.6.3.3 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.

##### 3.3.6.3.4 Notes on SIF\_EncryptionLevel

**Export Limitations**

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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "EXPORT).

**Cipher Algorithms**

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 3.3.6.3.4-1: Key Lengths

For more information regarding this topic, please refer to Chapter 7 of [[Schneier]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "Schneier).

#### 3.3.6.4 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.

#### 3.3.6.5 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 ZISs 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.

#### 3.3.6.6 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.



Figure 19: Push Model

"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.



Figure 20: Pull Model

"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.

##### 3.3.6.6.1 The "Push" Model



Figure 21: Push Mode

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.

##### 3.3.6.6.2 The "Pull" Model



Figure 22: Get Message in Pull Mode

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\_Status>

</SIF\_Ack>

</SIF\_Message>

Example 3.3.6.6.2-1: 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.

1. Typically a pull-mode agent removes the message from its queue by sending an "Immediate" SIF\_Ack to the ZIS;
2. 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.
3. 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 3.3.6.6.2-2: The "Pull" Model - SIF\_Status/SIF\_Code of 9

**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.

##### 3.3.6.6.3 "Multiple Version" Zones

It is possible for a zone to contain agents written to different versions of the SIF Implementation Specification. 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.

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



Figure 23: Infrastructure Transport Layer

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.

#### 3.3.7.1 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2616) 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]](http://specification.sifinfo.org/Implementation/2.4/References.html#RFC2246); however, SSL 3.0 [[SSL3]](http://specification.sifinfo.org/Implementation/2.4/References.html#SSL3) is also supported and SSL 2.0 client hellos [[SSL2]](http://specification.sifinfo.org/Implementation/2.4/References.html#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]](http://specification.sifinfo.org/Implementation/2.4/References.html#RFC2616) 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 (Section 3.6 of [[RFC 2616]](http://specification.sifinfo.org/Implementation/2.4/References.html#RFC2616)) 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.

##### 3.3.7.1.1 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.

**UTF-8 Encoding**

##### 3.3.7.1.2 HTTP Request Headers

The following HTTP request and common headers defined in [[RFC 2616]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2616)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 3.3.7.1.2-1: HTTP Request Headers

Note that all header values MUST conform to the requirements of [[RFC 2616]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2616) 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 3.3.7.1.2-1: 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.

##### 3.3.7.1.3 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 3.3.7.1.3-1: HTTP Response Headers

Note that all header values MUST conform to the requirements of [[RFC 2616]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2616) 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 3.3.7.1.3-1: 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.

##### 3.3.7.1.4 100 (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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2616) 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]](http://specification.sifinfo.org/Implementation/2.4/References.html#RFC2616) specification, possibly returning an intermediate response with 100 (Continue) status, for communication to proceed correctly.

##### 3.3.7.1.5 3XX, 4XX, 5XX Notices

A server should return 200-OK response notices but may return other notices. Servers built using existing web server technology are more likely to 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.

#### 3.3.7.2 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.

#### 3.3.7.3 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2616) 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 3.3.7.3-1: 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 3.3.7.3-2: 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 3.3.7.3-3: 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 3.3.7.3-4: 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.

#### 3.3.7.4 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "RFC2616). |

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 3.3.7.4-1: 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 3.3.7.4-2: 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.

#### 3.3.7.5 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](http://specification.sifinfo.org/Implementation/2.4/Architecture.html" \l "SIFHTTPSTransport) or [SIF HTTP Transport](http://specification.sifinfo.org/Implementation/2.4/Architecture.html#SIFHTTPTransport) 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.

#### 3.3.7.6 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.

#### 3.3.7.7 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.

#### 3.3.7.8 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\_Messages. 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.

## 4.1 Agent Protocols

### 4.1.1 Agent Messaging Protocols

This section documents how Agents should send and receive individual messages and the Agent’s resulting action.

#### 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 24: SIF\_Register flowchart

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Register](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Register) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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 25: SIF\_Unregister

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Unregister](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Unregister) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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 removed from the Zone. | Messaging protocol complete (success). |
| 5 | Messaging protocol has failed due to a [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Provide) message to the ZIS. To do this 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 adding new objects with a SIF\_Provide that the Agent is still the Provider of any objects for which it was previously registered as the Provider. Use SIF\_Unprovide to unregister as the Provider of any given objects. Use SIF\_Provision to replace all objects your Agent provides in one operation,.

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 26: SIF\_Provider flowchart

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Provide](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Provide) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 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's support for SIF\_ExtendedQuery can be specified in SIF\_ExtendedQuerySupport for each object. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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 the Provider of each of the objects specified in the SIF\_Provide message, in the applicable context(s). Any request sent by an Agent for one of these objects without explicitly specifying a particular Responder in SIF\_Header/SIF\_DestinationId will be placed in your Agent's message queue. | Messaging protocol complete (success). |
| 5 | Messaging protocol has failed due to a [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Unprovide) message to the ZIS. Note that any SIF\_Requests 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 27: SIF\_Unprovide flowchart

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Unprovide](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Unprovide) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 provide, 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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 no longer the Provider of each of the objects specified in the SIF\_Unprovide message, in the applicable context(s). SIF\_Requests 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 [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Subscribe) message to the ZIS. To do this 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 the Agent is still subscribed to any objects to which it had previously subscribed. Use SIF\_Unsubscribe to unregister as a Subscriber of given objects. Use SIF\_Provision to replace all objects to which your Agent subscribes in one operation.

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 28: SIF\_Subscribe flowchart

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Subscribe](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Subscribe) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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 subscribed to each of the objects specified in the SIF\_Subscribe message, in the specified context(s) if included. Any SIF\_Events for these objects will be placed in your Agent's queue. | Messaging protocol complete (success). |
| 5 | Messaging protocol has failed due to a [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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 29: SIF\_Unsubscribe flowchart

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Unsubscribe](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Unsubscribe) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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. Itis an alternative to SIF\_Provide, SIF\_Unprovide, SIF\_Subscribe and SIF\_Unsubscribe.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 30: SIF\_Provision flowchart

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Provision](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Provision) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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.  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.  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).  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).  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.  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. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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 that object, including your Agent if it is a subscriber.



Figure 31: SIF Event

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Event](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Event) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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.  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.  Add Event  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 xsi:nil set to true if necessary.  Change Event  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 xsi:nil set to true if necessary. For each list of repeatable elements in the object that has changed, include the whole list if the list type indicated is List. If the list type is ActionList, 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 ActionList, 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 Delete on the deleted child element in the list. Omitting an element in an ActionList indicates that it has been unchanged in the event. Refer to the Data Model section of the specification, [Lists/Repeatable Elements](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#ListsRepeatableElements), for more details on ActionLists and Lists.  Delete Event  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. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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 32: SIF\_Request

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Request](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Request) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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 33: SIF\_Ping

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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\_Ping](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ping) element in SIF\_SystemControlData. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) present? | If yes, go to Step 8. |
| 3 | Is SIF\_Status/SIF\_Code 0? | If no, go to Step 5. |
| 4 | The ZIS is awake. | Messaging protocol complete (success). |
| 5 | Is SIF\_Status/SIF\_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 [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) of 7 (your Agent sent a duplicate SIF\_MsgId). | Messaging protocol complete (failure). |
| 8 | Messaging protocol has failed due to a [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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". To wake up the push-mode Agent can either send a SIF\_Wakeup message or re-register with SIF\_Register.

A Pull-mode Agent can also changes it 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 pull-mode Agent is "awake," as will re-registering with SIF\_Register.



Figure 34: SIF\_Sleep

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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\_Sleep](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Sleep) element in SIF\_SystemControlData. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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 "asleep" in the ZIS. This is reflected to other Agents in SIF\_ZoneStatus and if your Agent is a Push-mode Agent, the ZIS will stop delivering messages to your Agent. To "wake up," send a SIF\_Wakeup message, or re-register with SIF\_Register. Pull-mode Agents may also send SIF\_GetMessage. | Messaging protocol complete (success). |
| 5 | Messaging protocol has failed due to a [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Wakeup) message to the ZIS to change the Agent’s state to "awake," whether sleeping or not; this state is available to other Agents via [SIF\_ZoneStatus](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_ZoneStatus). Upon success, the ZIS may begin delivering messages to a Push-mode Agent again, if previously sleeping.



Figure 35: SIF\_Wakeup

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Wakeup) element in SIF\_SystemControlData. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ZoneStatus)), send a [SIF\_GetZoneStatus](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_GetZoneStatus) message to the ZIS.



Figure 36: SIF\_GetZoneStatus

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_GetZoneStatus) element in SIF\_SystemControlData. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_AgentACL)), send a [SIF\_GetAgentACL](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_GetAgentACL) message to the ZIS.



Figure 37: SIF\_GetAgentACL

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_GetAgentACL) element in SIF\_SystemControlData. | Send SIF\_Message to ZIS over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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 38: SIF\_CancelRequests

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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\_CancelRequests](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_CancelRequests) element in SIF\_SystemControlData. |  |
| 2 | Specify Standard in NotificationType if your Agent desires or requires a "final" SIF\_Response be returned by the ZIS for each cancelled message (SIF\_Response/SIF\_MorePackets = No). Otherwise specify None. |  |
| 3 | Add a SIF\_RequestMsgIds element and add a child SIF\_RequestMsgId element for each SIF\_Request that the Agent wishes to cancel. | Send SIF\_Message to ZIS over appropriate transport. |
| 4 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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\_CancelRequests message. Your Agent will receive or not receive "final" SIF\_Responses per the specified NotificationType. | Messaging protocol complete (success). |
| 7 | Messaging protocol has failed due to a [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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 39: SIF\_GetMessage

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Message) in your agent's queue. | Messaging protocol complete (success). Process the returned SIF\_Message according to [Agent Message Handling Protocols](http://specification.sifinfo.org/Implementation/2.4/Messaging.html#AgentMessageHandlingProtocols) 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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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 40: SIF\_Ack Push Mode

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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 41: SIF\_Ack (Pull-Mode)

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Message/SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. | If your Agent is invoking SMB, go to Step 3. If your Agent is ending SMB, go to Step 4. |
| 2 | From the message being acknowledged/the message to be 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 1 (immediate SIF\_Ack) into SIF\_Code/SIF\_Data or an appropriate error description in SIF\_Error. If your Agent indicates a transport error or places 8 (receiver is sleeping) into SIF\_Code/SIF\_Data, the message will be acknowledged but remain in your Agent's queue. | Send SIF\_Message to ZIS over appropriate transport. Go to Step 5. |
| 3 | From the SIF\_Event being blocked, place the SIF\_Header/SIF\_SourceId value into SIF\_OriginalSourceId and place the SIF\_Header/SIF\_MsgId value into SIF\_OriginalMsgId. Place 2 (intermediate SIF\_Ack) into SIF\_Code/SIF\_Data. | Send SIF\_Message to ZIS over appropriate transport. Go to Step 5. |
| 4 | From the SIF\_Event being unblocked, 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. |
| 5 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) present? | If yes, go to Step 9. |
| 6 | Is SIF\_Status/SIF\_Code 0? | If no, go to Step 8. |
| 7 | If your Agent did not invoke or end SMB for a SIF\_Event, 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).  If your Agent invoked SMB by sending an intermediate SIF\_Ack, delivery of events is blocked until your Agent removes the SIF\_Event from its queue by sending a final SIF\_Ack. Your Agent will continue to receive SIF\_Responses and SIF\_Requests.  If your Agent ended SMB by sending a final SIF\_Ack, the ZIS has removed the blocked event from your Agent's queue and resumes delivery of events to your Agent. | Messaging protocol complete (success). |
| 8 | Messaging protocol has failed due to a [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) of 8 (ZIS is asleep) or 7 (your Agent sent a duplicate SIF\_MsgId). | Messaging protocol complete (failure). |
| 9 | Messaging protocol has failed due to a [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, 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 providing 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 relating to a SIF Zone service may only be transmitted by the agent registered in the zone as the default provider of the service. Unlike Event messages, SIF\_ServiceNotify messages may be packetized. This means a subcribing client must support the demuiltiplexing of simultaneously arriving service packets from multiple Zone Service Notification messages.



Figure 42: SIF\_ServiceNotify

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Initialize Current Packet Number to 1.  Initialize ServiceMsgId to a new UUID. The SIF\_ServiceMsgId MUST be the same for all packets. |  |
| 2 | Prepare a new SIF\_ServiceNotify message.  Initialize the SIF\_Header containing a new UUID in SIF\_MsgId, your Agent's Agent Id in SIF\_SourceId and the current time in SIF\_Timestamp.  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.  If this SIF\_ServiceNotify specifically applies to one or more contexts, place them in SIF\_Contexts; if omitted, the context is SIF\_Default.  Set SIF\_PacketNumber to the Current Packet Number.  Set SIF\_Service to the name of the SIF Zone Service.  Set SIF\_Operation to the name of the operation.  Set SIF\_ServiceMsgId to the ServiceMsgId created in Step 1. |  |
| 3 | Initialize SIF Zone Service operation SIF\_Body and set the appropriate values for the operation call.  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.  Add the SIF\_Body to the SIF\_ServiceNotify. | 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 |
| 8 | If a record could not be added to the SIF\_ServiceNotify:   * Set SIF\_MorePackets to No. * 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." * 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. * The agent should log the error. | 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 43: SIF\_ServiceInput

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Initialize Current Packet Number to 1  Initialize ServiceMsgId to a new UUID. The SIF\_ServiceMsgId MUST be the same for all packets. |  |
| 2 | Prepare a new SIF\_ServiceInput message.  Initialize the SIF\_Header containing a new UUID in SIF\_MsgId, your Agent's Agent Id in SIF\_SourceId and the current time in SIF\_Timestamp.  If your Agent would like to explicitly route this service operation to a given Agent, specify the Agent's Id in SIF\_DestinationId.  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.  Since a SIF\_ServiceInput applies only to the default context, it is not necessary to specify a value for SIF\_Contexts.  Set SIF\_PacketNumber to the Current Packet Number.  Set SIF\_Service to the name of the SIF Zone Service.  Set SIF\_Operation to the name of the operation.  Set SIF\_ServiceMsgId to the ServiceMsgId created in Step 1.  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.\*.  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. |  |
| 3 | Initialize SIF Zone Service operation SIF\_Body and set the appropriate values for the operation call.  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.  Add the SIF\_Body to the SIF\_ServiceInput. | 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 |
| 8 | If a record could not be added to the SIF\_ServiceInput:   * Set SIF\_MorePackets to No. * 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." * 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. * The agent should log the error. | 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 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 44: SIF\_CancelServiceInputs

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a SIF\_Message/SIF\_SystemControl message with SIF\_Header containing a new UUID 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.



Figure 45: SIF\_Message

| **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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. * If your Agent can scan the incoming message as UTF-8 encoded text to locate SIF\_Header/SIF\_SourceId and SIF\_Header/SIF\_MsgId, these values can be placed in SIF\_OriginalSourceId and SIF\_OriginalMsgId, respectively. * Otherwise include these elements with empty values, including an xsi:nil attribute value of true on SIF\_OriginalMsgId. * Include a SIF\_Error element with a SIF\_Category of 1 (XML Validation) and a SIF\_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 SIF\_Message? | If yes, go to step 5. |
| 4 | * Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 1 (XML Validation) and a SIF\_Code of 3 (generic validation error). | Go to step 12. |
| 5 | Is the namespace for SIF\_Message a namespace of a major version of SIF your Agent supports? Is SIF\_Message/@Version present with a value that your Agent supports? (If omitted, interpret SIF\_Message/@Version as 1.1.) | If yes, go to step 7. |
| 6 | * Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 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 SIF\_Message/@Version. Does the message validate? | If yes, go to step 9. |
| 8 | Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 1 (XML Validation) and an appropriate SIF\_Code from the corresponding choices in [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType). | Go to step 12. |
| 9 | If the namespace for SIF\_Message is for a previous major version of SIF, handle according to the specification for SIF\_Message/@Version.  Otherwise, is the message type (the child element of SIF\_Message) SIF\_Event, SIF\_Request, SIF\_Response, SIF\_Ping (Push-mode only), SIF\_Sleep (Push-mode only), SIF\_Wakeup (Push-mode only), or SIF\_CancelRequests (Push-mode only and your Agent chooses to support this optional message) | If yes, go to step 11. |
| 10 | Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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)](http://specification.sifinfo.org/Implementation/2.4/Messaging.html#SIF_AckPullMode) above. | Message handling is complete. |

Table 4.1.2.1-1: SIF\_Message Handling

#### 4.1.2.2 SIF\_Event

When a ZIS receives an event, it places the event in the queues of the Agents which have registered to receive this event. A SIF\_Event is delivered when it is at the top of 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.



Figure 46: SIF\_Event

| **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 | * Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Header) containing a new UUID 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. * If your Agent is a Push-Mode Agent, * Return the SIF\_Ack to the ZIS * 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. * If your Agent is a Pull-Mode Agent, * Send the SIF\_Ack to the ZIS per SIF\_Ack (Pull-Mode) above * 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. * If an error occurs, it is RECOMMENDED that your Agent publish a SIF\_LogEntry Add event. | Message handling complete. |
| 3 | * 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 SIF\_Ack (Pull-Mode) above. * 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. * Choose an option and process the event according to your Agent's business rules. When acknowledging: * Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Header) containing a new UUID 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, * 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. * If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a SIF\_LogEntry Add event. | 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

1. When an Agent sends a request directly to your Agent, or
2. 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
3. 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 SIF\_Header/SIF\_Contexts to determine the context for the request; if none is specified, the context is SIF\_Default. | Go to Step 3 if the context is supported. |
| 2 | * Prepare a SIF\_Response message with a copy of SIF\_Contexts, SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. * Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate General Message Handling and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the requested context is not supported. * Add SIF\_PacketNumber with a value of 1 and set SIF\_MorePackets to No. * Send the SIF\_Response to the original requester and acknowledge the error to the ZIS. | Message handling complete. |
| 3 | Examine the SIF\_Version element or elements specified in the SIF\_Request message. If more than one version is supported, select the highest version number supported. | Go to Step 5 if a version is supported. |
| 4 | * Prepare a SIF\_Response message with a copy of SIF\_Contexts, SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. * Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the requested SIF\_Versions are not supported. * Add SIF\_PacketNumber with a value of 1 and set SIF\_MorePackets to No. * Send the SIF\_Response to the original requester and acknowledge the error to the ZIS. | Message handling complete. |
| 5 | Examine the SIF\_MaxBufferSize specified in the SIF\_Request 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 | * Using the SIF version selected in Step 1, prepare a SIF\_Response message with SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. * Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the SIF\_MaxBufferSize cannot be honored. * Add SIF\_PacketNumber with a value of 1 and set SIF\_MorePackets to No. * Send the SIF\_Response to the original requester and acknowledge the error to the ZIS. | Message handling complete. |
| 7 | Is SIF\_ExtendedQuery specified? | If yes, go to Step 12. |
| 8 | The query type is SIF\_Query. Examine the object name being queried in SIF\_QueryObject/@ObjectName. | Go to Step 10 if the object is supported. |
| 9 | * Prepare a SIF\_Response message using the version chosen in Step 1 with SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. * Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the object is not supported. * Add SIF\_PacketNumber with a value of 1 and set SIF\_MorePackets to No. * Send the SIF\_Response to the original requester and acknowledge the error to the ZIS. | Stop processing the message. |
| 10 | Examine the query represented, if any, by SIF\_ConditionGroup, or SIF\_Example in the case of objects that support query-by-example, and determine whether it is supported. | Go to Step 14 if either SIF\_ConditionGroup or SIF\_Example is present, and if the query represented by SIF\_ConditionGroup or SIF\_Example is supported. |
| 11 | * Prepare a SIF\_Response message with SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. * Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the query is not supported. * Add SIF\_PacketNumber with a value of 1 and set SIF\_MorePackets to No. * Send the SIF\_Response to the original requester and acknowledge the error to the ZIS. | Stop processing the message. |
| 12 | Examine the query represented by SIF\_ExtendedQuery, and determine whether it is supported. | Go to Step 14 if the query is supported. |
| 13 | * Prepare a SIF\_Response message with SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. * Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the query is not supported. * Add SIF\_PacketNumber with a value of 1 and set SIF\_MorePackets to No. * Send the SIF\_Response to the original requester and acknowledge the error to the ZIS. | Stop processing the message. |
| 14 | * Note that Push-Mode Agents should acknowledge receipt of the SIF\_Request as response generation is typically a long-running operation that will typically lead to HTTP time-outs. Upon the time-out the request will be removed from the Agent's queue, therefore it is RECOMMENDED that the Push-Mode Agent persist the request and its SIF\_PacketNumber 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 SIF\_Error, SIF\_PacketNumber set to the last successfully transmitted SIF\_PacketNumber + 1 and SIF\_MorePackets set to No. * 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. * Pull-Mode Agents can choose to acknowledge receipt of the request here or at the end of response generation. * Initialize packet counter to 1. |  |
| 15 | Prepare a SIF\_Response message with SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. When handling SIF\_ExtendedQuery, copy the requested columns into SIF\_ExtendedQueryResults/SIF\_ColumnHeaders. |  |
| 16 | Add one or more of the matching objects into SIF\_ObjectDatafor SIF\_Query, or rows into SIF\_ExtendedQueryResult, for SIF\_ExtendedQuery, until no more will fit within the specified buffer size.  If no objects or rows will fit within SIF\_MaxBufferSize, go to Step 15 with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that SIF\_MaxBufferSize cannot be honored. Otherwise, note that only requested columns are returned when processing SIF\_ExtendedQuery. When processing SIF\_Query, 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 SIF\_PacketNumber to the current packet counter and SIF\_MorePackets to No. Add an appropriate SIF\_Error element to the SIF\_Response and send the SIF\_Response to the ZIS. | Go to Step 21. |
| 18 | Set SIF\_PacketNumber to the current packet counter value and set SIF\_MorePackets appropriately. Send the SIF\_Response to the ZIS. |  |
| 19 | Examine the SIF\_Ack returned by the ZIS. | If an error occurred, stop processing the SIF\_Request 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

When a responder sends a response to an Agent’s request, the ZIS places the SIF\_Response on the queue of the requesting Agent.

It is delivered when it is the next message available for delivery to the requesting Agent.



Figure 47: SIF\_Response

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | SIF\_RequestMsgId indicates which of your SIF\_Requests this packet is in response to. Is SIF\_Error present? | If no, go to Step 3. |
| 2 | * The Responder's handling of your Agent's SIF\_Request has failed due to a [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html" \l "InfrastructureErrorCategoryType) 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. * If your Agent is a Pull-Mode Agent, acknowledge the message per SIF\_Ack (Pull-Mode) above. * If your Agent is a Push-Mode Agent: Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Header) containing a new UUID 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, * return the SIF\_Ack to the ZIS. | Message handling complete. Any resources associated with the request can be released. |
| 3 | 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.  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.  Choose an option and process the response according to your Agent's business rules. When acknowledging: Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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.  If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a SIF\_LogEntry Add event. | 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. |

Table 4.1.2.4-1: SIF\_Response Handling

#### 4.1.2.5 SIF\_Ping (Push-Mode only)

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



Figure 48: SIF\_Ping (Push-Mode only)

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. If your Agent is "awake," include a SIF\_Status element with a SIF\_Code of 1 (immediate SIF\_Ack). Otherwise you may optionally notify the ZIS that your Agent is asleep by returning a SIF\_Code of 8 (receiver is sleeping). |  |
| 2 | Return the SIF\_Ack 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.



Figure 49: SIF\_Sleep

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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.



Figure 50: SIF\_Wakeup

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 requests that your Agent cancela 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](http://specification.sifinfo.org/Implementation/2.4/Messaging.html" \l "AgentMessageHandlingSIF_Message) handling protocol.



Figure 51: SIF\_CancelRequests

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Messaging.html" \l "AgentMessageHandlingSIF_Message) handling protocol.



Figure 52: SIF\_CancelServiceInputs

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | Prepare a [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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\_ServiceOutput packets for any of the SIF\_ServiceInput messages specified in the SIF\_ServiceMsgId element(s), stop processing the request(s). If your Agent has any of the specified SIF\_ServiceInput queued locally, remove them from the agent local queue. |  |
| 3 | Return the SIF\_Ack to the ZIS. | Message processing complete (success). |

Table 4.1.2.9-1: 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 on 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.

SIF\_ServiceOutput can apply to only one context. Therefore 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.



Figure 53: SIF\_ServiceNotify

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | 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.  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.  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 UUID 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.  If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a SIF\_LogEntry Add event. | 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 exposed by a SIF Zone Service.

A ZIS places a SIF\_ServiceInput in your Agent's queue when another Agent sends a directed service request to your Agent, or when another 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. |  |
| 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 |  |
| 12 | Initialize Current Packet Number to 1 Collect sender's SIF\_SourceId from SIF\_ServiceInput/SIF\_Header/SIF\_SourceId Collect Maximum Buffer Size from SIF\_ServiceInput/SIF\_MaxBufferSize Collect SIF\_Version(s) from SIF\_ServiceInput/SIF\_Version |  |
| 13 | Prepare a new SIF\_ServiceOutput message. Initialize the SIF\_Header containing a new UUID in SIF\_MsgId, your Agent's Agent Id in SIF\_SourceId and the current time in SIF\_Timestamp. 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.  Since SIF\_ServiceOutput can apply to only one context, the value in SIF\_Contexts is always SIF\_Default. Set SIF\_ServiceOutput/SIF\_Header/SIF\_DestinationId to the value from SIF\_ServiceInput/SIF\_Header/SIF\_SourceId Set SIF\_PacketNumber to the Current Packet Number. Set SIF\_Service to the name of the SIF Zone Service. Set SIF\_Operation to the name of the operation. Set SIF\_ServiceMsgId to the SIF\_ServiceMsgId of the original SIF\_ServiceInput/SIF\_ServiceMsgId |  |
| 14 | Initialize SIF Zone Service operation SIF\_Body and set the appropriate values for the operation call. 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. Add the SIF\_Body to the SIF\_ServiceOutput | 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 | If a record could not be added to the SIF\_ServiceOutput Set SIF\_MorePackets to No Create a new SIF\_Error with the SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc set appropriately. 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. The agent should log the error. | 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.



Figure 54: SIF\_ServiceOutput

| **Step** | **Process** | **Flow Control** |
| --- | --- | --- |
| 1 | SIF\_ServiceInMsgId indicates which of your SIF\_ServiceInputs this packet is in response to. Is SIF\_Error present? | If no, go to Step 3. |
| 2 | The Responder's handling of your Agent's SIF\_ServiceInput 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.  If your Agent is a Pull-Mode Agent, acknowledge the message per SIF\_Ack (Pull-Mode) above.  If your Agent is a Push-Mode Agent: Prepare a SIF\_Ack message with SIF\_Header containing a new UUID 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. | Message handling complete. Any resources associated with the request can be released. |
| 3 | 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.  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.  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 UUID 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.  If an error occurs regardless of the option chosen, it is RECOMMENDED that your Agent publish a SIF\_LogEntry Add event. | 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. |

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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. |
| 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 SIF\_Status/SIF\_Code 2 (intermediate SIF\_Ack)? | If no, go to Step 23. |
| 20 | The Agent is invoking SMB. Is the delivered message a SIF\_Event? | 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 SIF\_LogEntry  Add event with the appropriate error category of 13 (SMB Error) and code 2 (SMB can only be invoked for SIF\_Event), containing a copy of the SIF\_Header of the queued message. SIF\_LogEntry/SIF\_Desc MUST contain the SIF\_SourceId 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 SIF\_Event. Persist that the Agent has invoked SMB along with the SIF\_MsgId 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 SIF\_Ack with this SIF\_MsgId in SIF\_OriginalMsgId, or by sending a SIF\_Wakeup 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 SIF\_Status/SIF\_Code 8 (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 [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) of 7 (already have this SIF\_MsgId). The ZIS cannot correct this, as the SIF\_MsgId 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 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 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 [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, if included. If SIF\_Category 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 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 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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 [SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID in SIF\_MsgId, the Zone Id in SIF\_SourceId and the current time in SIF\_Timestamp; other SIF\_Header elements do not apply. Place an empty [SIF\_Ping](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ping) element in SIF\_SystemControlData. | Send SIF\_Message to Agent over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) present? | If yes, go to Step 8. |
| 3 | Is SIF\_Status/SIF\_Code 1? | If no, go to Step 5. |
| 4 | The Agent is awake. | Messaging protocol complete (success). |
| 5 | Is SIF\_Status/SIF\_Code 8 (receiver is sleeping)? | If no, go to Step 7. |
| 6 | The Agent is asleep. | Messaging protocol complete (success). |
| 7 | Messaging protocol has failed due to a [SIF\_Status/SIF\_Code](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) of 7 (your ZIS sent a duplicate SIF\_MsgId). | Messaging protocol complete (failure). |
| 8 | Messaging protocol has failed due to a [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Sleep) element in SIF\_SystemControlData. | Send SIF\_Message to Agent over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Wakeup) element in SIF\_SystemControlData. | Send SIF\_Message to Agent over appropriate transport. |
| 2 | Receive [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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](http://specification.sifinfo.org/Implementation/2.4/Messaging.html#ZISMessageHandlingSIF_CancelRequests). 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "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](http://specification.sifinfo.org/Implementation/2.4/Messaging.html#ZISMessageHandlingSIF_CancelServiceInputs) . 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ack) in response. Is [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) 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\_CancelServiceInputs. | 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 Version attribute of SIF\_Message 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 SIF\_Ack containing a SIF\_Error element. (Note that if XML is not well-formed, or invalid and the well-formed XML is not made available by the XML parser, SIF\_SourceId and SIF\_MsgId will not be available from the incoming XML message. If this is the case, include SIF\_OriginalSourceId and SIF\_OriginalMsgId in the SIF\_Ack as empty elements with xsi:nil set to true as necessary to indicate the current message.) Set SIF\_Error/SIF\_Category to indicate XML Validation and place the appropriate error code and description in SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc. Place any additional parser information into SIF\_Error/SIF\_ExtendedDesc. Return the SIF\_Ack to caller. If it can be determined the message is a SIF\_Response, see SIF\_Response Handling below, Step 13, to send an error SIF\_Response to the requester. | Stop processing this message. |
| 3 | Examine the Version attribute of the message. | If the version is supported, go to Step 5. |
| 4 | Prepare a SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to Generic Message Handling, indicating that the message is not supported in SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc. Return the SIF\_Ack to the caller. If this message is a SIF\_Response, see SIF\_Response Handling below, Step 13, to send an error SIF\_Response to the requester. | Stop processing this message. |
| 5 | Examine message's SIF\_Header to retrieve the SIF\_SourceId and the message to get the message type. If message type is not SIF\_Register, determine if the sender identified by SIF\_SourceId is registered. | If message type is SIF\_Register or if the sender's SIF\_SourceId is registered, go to Step 9. |
| 6 | Prepare a SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to Access and Permissions, indicating that the sender is not registered in SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc. Return the SIF\_Ack to the caller. If this message is a SIF\_Response, see SIF\_Response Handling below, Step 13, to send an error SIF\_Response 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. |
| 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 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 SIF\_SourceId of the message. The ZIS must remove the agent from its list of registered agents. Perform SIF\_Unprovide functionality for any objects the agent is providing. Perform SIF\_Unsubscribe functionality for any objects to which the agent is subscribed. Discard any pending messages for the agent. |  |
| 2 | Prepare a SIF\_Ack containing a SIF\_Status element indicating success. Return the SIF\_Ack 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 SIF\_Ack. | 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 SIF\_ZoneStatus). | If object name is valid, go to Step 5. |
| 4 | Add a SIF\_Error element to the SIF\_Ack. Set SIF\_Error/SIF\_Category to indicate Provision and set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the object is invalid. Place the name of the invalid object in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 14. |
| 5 | If no SIF\_Context is specified, the context is SIF\_Default. Otherwise check that each SIF\_Context supplied in SIF\_Contexts is supported. | If they are all supported, go to Step 7. |
| 6 | Prepare a SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Generic Message Handling. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate a context is not supported. Place the name of the context in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 14. |
| 7 | Using the SIF\_SourceId, 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 SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Access and Permissions. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the sender lacks permission to provide this object. Place the name of the object in SIF\_Error/SIF\_ExtendedDesc. | 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. |
| 10 | Is the current provider the same as the SIF\_SourceId of this message? | If the provider differs from the SIF\_SourceId of this message, go to Step 14. Otherwise go to Step 2. |
| 11 | Add a record in the Providers database to indicate that SIF\_SourceId is the provider of this object in the given contexts. If an error occurs, add a SIF\_Error element to the SIF\_Ack. | If an error occurs, go to Step 13; otherwise go to Step 2. |
| 12 | Add a SIF\_Error element to the SIF\_Ack. Set SIF\_Error/SIF\_Category to indicate Provision and set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the object already has a provider. Place the name of the provider in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 14. |
| 13 | Add a SIF\_Status element indicating success to the SIF\_Ack. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 14 | Undo all changes to the Providers database. Return the SIF\_Ack 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 SIF\_Ack. | 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 SIF\_ZoneStatus). | Go to Step 5 if the object name is valid. |
| 4 | Add a SIF\_Error element to the SIF\_Ack. Set SIF\_Error/SIF\_Category to indicate Provision and set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the object is invalid. Place the name of the invalid object in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 10. |
| 5 | If no SIF\_Context is specified, the context is SIF\_Default. Otherwise check that each SIF\_Context supplied in SIF\_Contexts is supported. | If they are all supported, go to Step 7. |
| 6 | Prepare a SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Generic Message Handling. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate a context is not supported. Place the name of the context in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 10. |
| 7 | If it exists, remove the records in the Providers database that marks SIF\_SourceId as the provider of this object for the given contexts. If an error occurs, add a SIF\_Error element to the SIF\_Ack. | If an error occurs, go to Step 10. |
| 8 | Leave all pending SIF\_Requests for the object in the responder's queue, as they may include SIF\_Requests routed explicitly to the responder using SIF\_DestinationId. | Go to Step 2. |
| 9 | Add a SIF\_Status element indicating success to the SIF\_Ack. Return the SIF\_Ack to the caller | Stop processing the message. |
| 10 | Undo all changes to the Providers database. Return the SIF\_Ack 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 SIF\_Ack. | 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 SIF\_Error element to the SIF\_Ack. Set SIF\_Error/SIF\_Category to indicate Subscription and set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the object is invalid. Place the name of the invalid object in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 12. |
| 5 | If no SIF\_Context is specified, the context is SIF\_Default. Otherwise check that each SIF\_Context supplied in SIF\_Contexts is supported. | If they are all supported, go to Step 7. |
| 6 | Prepare a SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Generic Message Handling. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate a context is not supported. Place the name of the context in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 12. |
| 7 | Using the SIF\_SourceId, 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 SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Access and Permissions. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the sender lacks permission to subscribe to this object. Place the name of the object in SIF\_Error/SIF\_ExtendedDesc. | 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 SIF\_SourceId is a subscriber of this object's SIF\_Events in the specified contexts. If an error occurs, add a SIF\_Error element to the SIF\_Ack. | If an error occurs go to Step 12; otherwise go to Step 2. |
| 11 | Add a SIF\_Status element indicating success to the SIF\_Ack. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 12 | Undo all changes to the Subscribers database. Return the SIF\_Ack 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 SIF\_Ack. | 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. |
| 4 | Add a SIF\_Error element to the SIF\_Ack. Set SIF\_Error/SIF\_Category to indicate Subscription and set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the object is invalid. Place the name of the invalid object in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 9. |
| 5 | If no SIF\_Context is specified, the context is SIF\_Default. Otherwise check that each SIF\_Context supplied in SIF\_Contexts is supported. | If they are all supported, go to Step 7. |
| 6 | Prepare a SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Generic Message Handling. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate a context is not supported. Place the name of the context in SIF\_Error/SIF\_ExtendedDesc. | Go to Step 12. |
| 7 | If it exists, remove the record in the Subscribers database that marks SIF\_SourceId as a subscriber of this object's SIF\_Events in the specified contexts. If an error occurs, add a SIF\_Error element to the SIF\_Ack. | If an error occurs go to Step 9, otherwise go to Step 2. |
| 8 | Add a SIF\_Status element indicating success to the SIF\_Ack. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 9 | Undo all changes to the Subscribers database. Return the SIF\_Ack 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 SIF\_Ack. |  |
| 2 | Process SIF\_ProvideObjects as provide. | On error go to step 13. |
| 3 | Process objects not in SIF\_ProvideObjects as unprovide. | On error go to step 13. |
| 4 | Process SIF\_SubscribeObjects as subscribe. | On error go to step 13. |
| 5 | Process objects not in SIF\_SubscribeObjects as unsubscribe. | On error go to step 13. |
| 6 | Process SIF\_PublishAddObjects. | On error go to step 13. |
| 7 | Process SIF\_PublishChangeObjects. | On error go to step 13. |
| 8 | Process SIF\_PublishDeleteObjects. | On error go to step 13. |
| 9 | Process SIF\_RequestObjects. | On error go to step 13. |
| 10 | Process SIF\_RespondObjects. | On error go to step 13. |
| 11 | Save changes. |  |
| 12 | Return success SIF\_Ack. | Stop processing. |
| 13 | Roll back any changes. |  |
| 14 | Return error SIF\_Ack. | 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. |
| 2 | Add a SIF\_Error element to the SIF\_Ack. Set SIF\_Error/SIF\_Category to indicate Event Reporting and set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the event is invalid. Place the name of the invalid object in SIF\_Error/SIF\_ExtendedDesc. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 3 | If no SIF\_Context is specified, the context is SIF\_Default. Otherwise check that each SIF\_Context supplied in SIF\_Contexts is supported. | If they are all supported, go to Step 5. |
| 4 | Prepare a SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Generic Message Handling. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate a context is not supported. Place the name of the context in SIF\_Error/SIF\_ExtendedDesc. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 5 | Using the SIF\_SourceId, 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 SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Access and Permissions. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the sender lacks permission to publish events pertaining to this object (use general SIF\_Event error code, or specific Add, Change, Delete codes). Place the name of the object in SIF\_Error/SIF\_ExtendedDesc. Return the SIF\_Ack 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 SIF\_Event. | Go to Step 9 if there are no subscribers for this object. |
| 8 | For each subscriber make a copy of the SIF\_Event. If the ZIS supports XML filtering pass the copy to the XML filter logic. If an XML filter matched the root SIF\_Message 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 SIF\_Event, 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. |

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 SIF\_Ack. |  |
| 2 | Retrieve the name of the object from the ObjectName attribute of SIF\_Query/SIF\_QueryObject and check whether it's a valid/supported object. | Go to Step 4 if the object name is valid. |
| 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 object name is invalid. Place the name of the invalid object in SIF\_Error/SIF\_ExtendedDesc. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 4 | If no SIF\_Context is specified, the context is SIF\_Default. Otherwise check that the context supplied in SIF\_Contexts 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 SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Generic Message Handling. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc 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 SIF\_Error/SIF\_ExtendedDesc. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 6 | Using the SIF\_SourceId, 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 SIF\_Ack containing a SIF\_Error element. Set SIF\_Error/SIF\_Category to indicate Access and Permissions. Set SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the sender lacks permission to request this object. Place the name of the object in SIF\_Error/SIF\_ExtendedDesc. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 8 | Examine the SIF\_Request header looking for a SIF\_DestinationId | Go to Step 11 if a SIF\_DestinationId was located. |
| 9 | No SIF\_DestinationId 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 SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response 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. |
| 11 | A SIF\_DestinationId was specified indicating the responder. Confirm that the agent specified in SIF\_DestinationId has permission to send SIF\_Response 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 SIF\_Provision and/or SIF\_Provide that the Responder cannot handle a SIF\_Query for a given object or SIF\_ExtendedQuery for any referenced object, or that the Responder doesn't handle extended queries in general, add a SIF\_Error element with the applicable SIF\_Error/SIF\_Category and SIF\_Error/SIF\_Code (object not supported, query not supported, or SIF\_ExtendedQuery not supported). Place an appropriate error message in SIF\_Desc and/or SIF\_ExtendedDesc. Return the SIF\_Ack to the caller. | Stop processing the message. |
| 13 | If the ZIS supports SIF XML filter pass apply the SIF XML filter logic to the SIF\_Request. If a rule applied to the root element SIF\_Message then Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate that the SIF\_Request was canceled due to a SIF XML filter rule. Return the SIF\_Ack to the caller. | Stop processing the message if an error SIF\_Ack was returned. |
| 14 | Deposit the SIF\_Request 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\_Request, 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 request. |  |
| 15 | Return a SIF\_Ack, with SIF\_Status set to 0, to the caller to indicate that SIF\_Request 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. |
| 14 | Prepare a SIF\_Response message with SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message.  Add a SIF\_Error element with the SIF\_Error/SIF\_Category set to indicate Request and Response and SIF\_Error/SIF\_Code and SIF\_Error/SIF\_Desc to indicate the reason that the SIF\_Response packet was rejected.  Add SIF\_PacketNumber with a value set to set to 1 + the previous SIF\_PacketNumber and SIF\_MorePackets to No.  Send the SIF\_Response to the original requester. 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\_Response validation. | Stop processing the message.  The ZIS must also guarantee that no additional SIF\_Response packets for this SIF\_Request will be accepted. Depending on the implementation, the ZIS may need to alter the SIF\_Request cache it maintains to signal that the SIF\_Request is no longer valid.  The ZIS may remove the SIF\_Request from the cache as the stream is closed. |

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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 your ZIS is "awake," include a SIF\_Status element with a SIF\_Code of 0 (success). Otherwise you may optionally notify the Agent that your ZIS is asleep by returning a SIF\_Code of 8 (receiver is sleeping). |  |
| 2 | Return the SIF\_Ack 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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. Include a SIF\_Status element with a SIF\_Code of 0 (success). Change the state of the Agent to "asleep." |  |
| 2 | Return the SIF\_Ack 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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. Include a SIF\_Status element with a SIF\_Code of 0 (success). Change the Agent's state to "awake." |  |
| 2 | Return the SIF\_Ack 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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. Include a SIF\_Status element with a SIF\_Code of 0 (success). Reflect the current state of the zone in SIF\_Status/SIF\_Data/SIF\_ZoneStatus. |  |
| 2 | Return the SIF\_Ack 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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. Include a SIF\_Status element with a SIF\_Code of 0 (success). Communicate the Agent's ACL permissions in SIF\_Status/SIF\_Data/SIF\_AgentACL. |  |
| 2 | Return the SIF\_Ack 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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. Include a SIF\_Status element with a SIF\_Code of 0 (success). Return the SIF\_Ack to the Agent. (There are no error return values that apply to this message.) | Go to Step 2. |
| 2 | For each SIF\_RequestMsgId element, perform the following steps. | If all SIF\_RequestMsgId elements have been processed, processing is complete. |
| 3 | Using the supplied SIF\_RequestMsgId, look up the SIF\_Request that initiated this response. | Go to Step 2 if the SIF\_Request is not found, or has already been completed with a "final" SIF\_Response packet (SIF\_MorePackets = No). |
| 4 | Examine the SIF\_SourceId specified in the SIF\_Request message and compare it to the SIF\_SourceId in the SIF\_SystemControl message. | If the SIF\_SourceId is not the same, go to Step 2. |
| 5 | Close out the SIF\_Request 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 SIF\_CancelRequests message to that Agent. (Note: This could also be accomplished by packaging up all SIF\_RequestMsgIds that apply to the same responding Agent and sending a single SIF\_CancelRequests message.) |  |
| 7 | Examine the value of SIF\_NotificationType. | If set to Standard, go to Step 8. If set to None, go to Step 10. |
| 8 | Prepare a SIF\_Response message with SIF\_DestinationId set to SIF\_SourceId and SIF\_RequestMsgId set to SIF\_MsgId from the SIF\_Request message. |  |
| 9 | Add a SIF\_Error element with the SIF\_Category set to indicate Request and Response, with SIF\_Code and SIF\_Desc indicating 18 (SIF\_Request cancelled by requesting agent).  Add SIF\_PacketNumber with a value set to the previous SIF\_PacketNumber + 1. Set SIF\_MorePackets to No.  Place the SIF\_Response in the requester's queue. |  |
| 10 | Determine if there are any more SIF\_RequestMsgId elements left to process. | Go to Step 2 if there are more SIF\_RequestMsgId 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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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. Include a SIF\_Status element with a SIF\_Code of 0 (success). Return the SIF\_Ack to the Agent. (There are no error return values that apply to this message.) | Go to Step 2. |
| 2 | For each SIF\_ServiceMsgId element, perform the following steps. | If all SIF\_ServiceMsgId elements have been processed, processing is complete. |
| 3 | Using the supplied SIF\_ServiceMsgId, look up the SIF\_ServiceInput that initiated this response. | Go to Step 2 if the SIF\_ServiceInput is not found, or has already been completed with a "final" SIF\_ServiceOutput packet (SIF\_MorePackets = No). |
| 4 | Examine the SIF\_SourceId specified in the SIF\_ServiceInput message and compare it to the SIF\_SourceId in the SIF\_SystemControl message. | If the SIF\_SourceId is not the same, go to Step 2. |
| 5 | Close out the SIF\_ServiceInput 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 SIF\_CancelServiceInputs message to that Agent. (Note: This could also be accomplished by packaging up all SIF\_ServiceMsgIds that apply to the same responding Agent and sending a single SIF\_CancelServiceInputs message.) |  |
| 7 | Examine the value of SIF\_NotificationType. | If set to Standard, go to Step 8. If set to None, go to Step 10. |
| 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. |  |
| 9 | Add a SIF\_Error element with the SIF\_Category set to indicate SIF Zone Service, with SIF\_Code and SIF\_Desc indicating 15 (SIF\_ServiceInput cancelled by requesting agent).  Add SIF\_PacketNumber with a value set to the previous SIF\_PacketNumber + 1. Set SIF\_MorePackets to No.  Place the SIF\_ServiceOutput in the requester's queue. |  |
| 10 | Determine if there are any more SIF\_ServiceMsgId elements left to process. | Go to Step 2 if there are more SIF\_ServiceMsgId 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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 | 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).  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. | 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 | 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).  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. | 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 [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) message with [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) containing a new UUID 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\_Error present? | If yes, go to Step 14. |
| 3 | Is SIF\_Status/SIF\_Code 1 (immediate SIF\_Ack)? | If no, go to Step 5. |
| 4 | If no message matches SIF\_OriginalMsgId, set SIF\_Error/SIF\_Category to 12 (Generic Message Handling) and SIF\_Error/SIF\_Code to 6 (no such message). Otherwise remove the identified message from the Agent's queue and set SIF\_Status/SIF\_Code to 0. | Return SIF\_Ack. Message handling complete. |
| 5 | Is SIF\_Status/SIF\_Code 2 (intermediate SIF\_Ack)? | If no, go to Step 7. |
| 6 | If no message matches SIF\_OriginalMsgId, set SIF\_Error/SIF\_Category to 12 (Generic Message Handling) and SIF\_Error/SIF\_Code to 6 (no such message). If the message identified is not a SIF\_Event, set SIF\_Category to 13 (SMB Error) and SIF\_Code to 2 (SMB can only be invoked on a SIF\_Event). Otherwise invoke SMB on the identified SIF\_Event, persisting SIF\_OriginalMsgId, and set SIF\_Status/SIF\_Code to 0. This event is blocked and all SIF\_Events are frozen. | Return SIF\_Ack. Message handling complete. |
| 7 | Is SIF\_Status/SIF\_Code 3 (final SIF\_Ack)? | If no, go to Step 9. |
| 8 | If SMB has not been invoked or the message identified by SIF\_OriginalMsgId doesn't exist or doesn't match the SIF\_Event blocked by SMB, set SIF\_Error/SIF\_Category to 13 (SMB Error) and SIF\_Error/SIF\_Code to 4 (incorrect SIF\_MsgId in final SIF\_Ack). (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 SIF\_Status/SIF\_Code to 0. | Return SIF\_Ack. Message handling complete. |
| 9 | Is SIF\_Status/SIF\_Code 7 (already have this SIF\_MsgId from you)? | If no, go to Step 11. |
| 10 | The ZIS cannot correct this, as the SIF\_MsgId originates from an Agent and can't be changed without other repercussions. Remove the message from the Agent's queue. Set SIF\_Status/SIF\_Code to 0. | Return SIF\_Ack. Message handling complete. |
| 11 | Is SIF\_Status/SIF\_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 SIF\_Status/SIF\_Code to 0. | Return SIF\_Ack. Message handling complete. |
| 13 | The Agent has violated protocol. Indicate SIF\_Error/SIF\_Category of 12 (Generic Message Handling Error) and SIF\_Error/SIF\_Code 5 (protocol error) for the message. | Return SIF\_Ack. Message handling complete. |
| 14 | The Agent has indicated a [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) condition. See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) with SIF\_Category and SIF\_Code, and examine SIF\_Desc and SIF\_ExtendedDesc, if included. If SIF\_Category does not indicate a transport error, remove the message from the Agent's queue. Otherwise it remains the next message to be delivered. Set SIF\_Status/SIF\_Code to 0. | Return SIF\_Ack. 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. |
| 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 | Prepare a SIF\_ServiceNotify message with a copy of the information in the original SIF\_ServiceNotify except for the SIF\_Body.  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.  Add SIF\_PacketNumber with a value set to set to 1 + the previous SIF\_PacketNumber and SIF\_MorePackets to No.  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. | Stop processing the message.  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.  The ZIS may remove the SIF\_ServiceNotify/SIF\_ServiceMsgId from the cache as the stream is closed. |

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. |
| 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. |
| 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 | Prepare a SIF\_ServiceInput message with a copy of the information in the original SIF\_ServiceInput except for the SIF\_Body.  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.  Add SIF\_PacketNumber with a value set to set to 1 + the previous SIF\_PacketNumber and SIF\_MorePackets to No.  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. | Stop processing the message.  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.  The ZIS may remove the SIF\_ServiceInput/SIF\_ServiceMsgId from the cache as the stream is closed. |

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. |
| 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. 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. | If the validation 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 "Packet number invalid, Buffer size exceeded." | If the packet stream has not been terminated by the ZIS go to step 9. |
| 5 | Examine the SIF\_ServiceInput header looking for a SIF\_DestinationId. Verify the SIF\_DestinationId matches the SIF\_SourceId in the original SIF\_ServiceInput. | Go to Step 7 if a SIF\_DestinationId was located. |
| 6 | 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. | Stop processing the message. |
| 7 | 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. | If the message cannot be put into the agent's queue and the SIF\_PacketNumber is greater than 1 go to step 9 |
| 8 | Prepare a SIF\_Ack containing a SIF\_Status element indicating success. Return a SIF\_Ack to the caller. | Stop processing the message. |
| 9 | Prepare a SIF\_ServiceOutput message with a copy of the information in the original SIF\_ServiceOutput except for the SIF\_Body.  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.  Add SIF\_PacketNumber with a value set to set to 1 + the previous SIF\_PacketNumber and SIF\_MorePackets to No.  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. | Stop processing the message.  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.  The ZIS may remove the SIF\_ServiceOutput/SIF\_ServiceMsgId from the cache as the stream is closed. |

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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html).

## 5.1 Common Elements

### 5.1.1 SIF\_Message

The SIF\_Message element is the root element of all SIF messages.

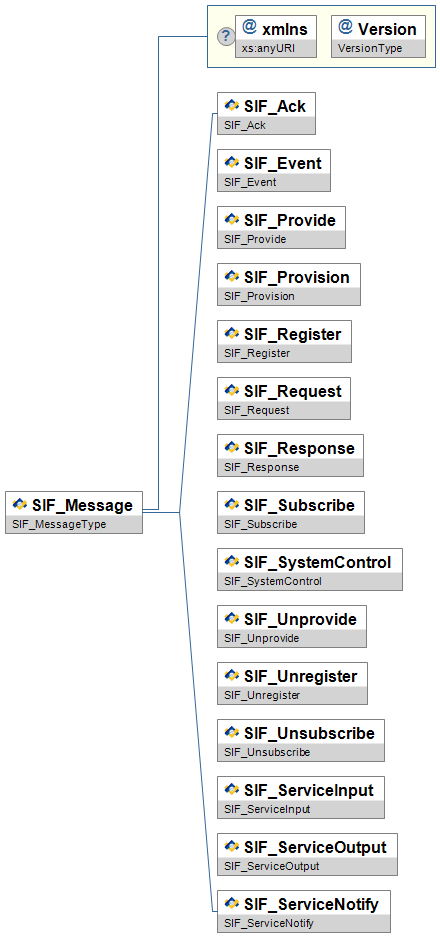
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Message.png)

Figure 5.1.1-1: SIF\_Message

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Message |  | Contains one of the SIF message types. | choice of:  [SIF\_Ack](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Ack) [SIF\_Event](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Event) [SIF\_Provide](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Provide) [SIF\_Provision](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Provision) [SIF\_Register](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Register) [SIF\_Request](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Request) [SIF\_Response](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Response) [SIF\_Subscribe](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Subscribe) [SIF\_SystemControl](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_SystemControl) [SIF\_Unprovide](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Unprovide) [SIF\_Unregister](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Unregister) [SIF\_Unsubscribe](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Unsubscribe) [SIF\_ServiceInput](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ServiceInput) [SIF\_ServiceOutput](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ServiceOutput) [SIF\_ServiceNotify](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ServiceNotify) |
| **@** | xmlns | C | The xmlns attribute specifies the XML namespace for SIF messages. For this version of the specification, the value of this attribute MUST be http://www.sifinfo.org/infrastructure/2.x . This XML namespace value will remain the same until the next major release of SIF (3.0).  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. | [xs:anyURI](http://www.w3.org/TR/xmlschema-2/#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 2.4. This attribute can be used by ZIS and agent implementations to choose schema files to validate the message's XML. | [VersionType](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#VersionType) |

Table 5.1.1-1: SIF\_Message

<SIF\_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">

...

</SIF\_Message>

Example 5.1.1-1: SIF\_Message

### 5.1.2 SIF\_Header

SIF\_Header is a common message header for all SIF messages.

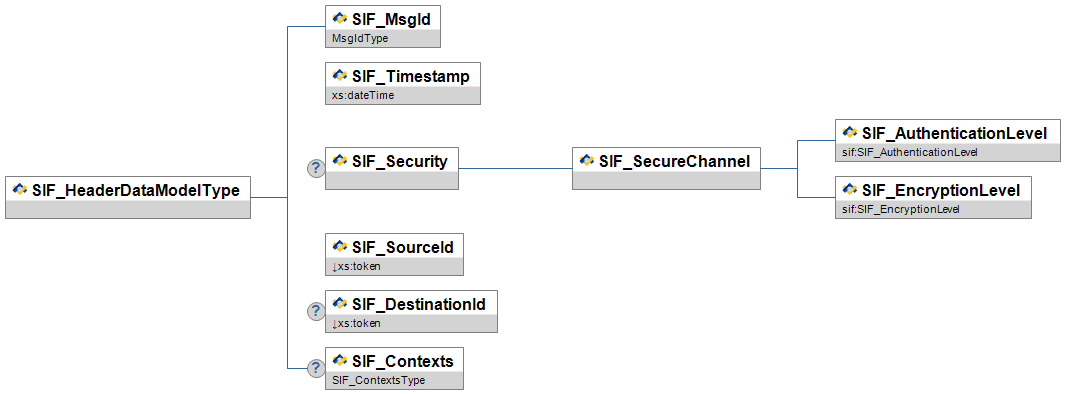
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Header.png)

Figure 5.1.2-1: 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](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#MsgIdType) |
|  | SIF\_Timestamp | M | Timestamp of when the message was sent. | [xs:dateTime](http://www.w3.org/TR/xmlschema-2/#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. |  |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_DestinationId | C | This element represents the Id of the recipient of the message and may be present as follows:  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.  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.  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.  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.  This element SHOULD NOT be used in any other SIF Infrastructure messages. If the element is present, it will be ignored by the ZIS. | [xs:token](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |

Table 5.1.2-1: 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 5.1.2-1: 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 5.1.2-2: SIF\_Header

### 5.1.3 SIF\_EncryptionLevel

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

[SIF_EncryptionLevel](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_EncryptionLevel.png)

Figure 5.1.3-1: 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 5.1.3-1: SIF\_EncryptionLevel

### 5.1.4 SIF\_AuthenticationLevel

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

[SIF_AuthenticationLevel](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_AuthenticationLevel.png)

Figure 5.1.4-1: 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 5.1.4-1: SIF\_AuthenticationLevel

### 5.1.5 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.

[SIF_Contexts](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Contexts.png)

Figure 5.1.5-1: 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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_Context | MR |  | [SIF\_Context](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Context) |

Table 5.1.5-1: SIF\_Contexts

### 5.1.6 SIF\_Context

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

[SIF_Context](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Context.png)

Figure 5.1.6-1: SIF\_Context

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Context |  | The name of a SIF Context that applies to a message or operation. | [xs:token](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |

Table 5.1.6-1: SIF\_Context

### 5.1.7 SIF\_Protocol

Contains protocol information regarding a ZIS or Agent.

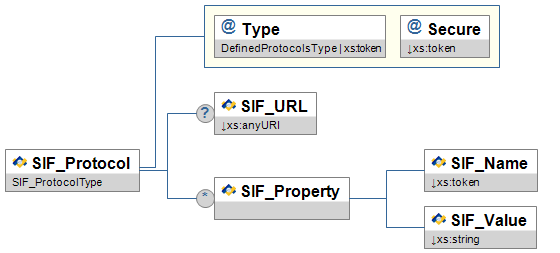
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Protocol.png)

Figure 5.1.7-1: 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:  [DefinedProtocolsType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#DefinedProtocolsType) [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
| **@** | 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. | [xs:anyURI](http://www.w3.org/TR/xmlschema-2/#anyURI)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |
|  | 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](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_Property/SIF\_Value | M | Property value. | [xs:string](http://www.w3.org/TR/xmlschema-2/#string)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |

Table 5.1.7-1: SIF\_Protocol

### 5.1.8 SIF\_Status

This element is used to signal a successful response.

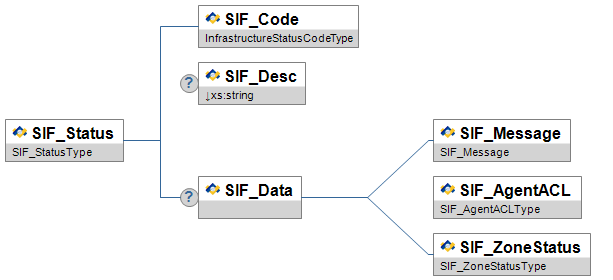
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Status.png)

Figure 5.1.8-1: SIF\_Status

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Status |  | This element is used to signal a successful response. |  |
|  | SIF\_Code | M |  | [InfrastructureStatusCodeType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureStatusCodeType) |
|  | SIF\_Desc | O | An optional textual description/equivalent of SIF\_Code. | [xs:string](http://www.w3.org/TR/xmlschema-2/#string)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 1024 | |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_Message) [SIF\_AgentACL](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_AgentACL) [SIF\_ZoneStatus](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ZoneStatus) |

Table 5.1.8-1: SIF\_Status

### 5.1.9 SIF\_Error

This element is used to signal an unsuccessful response.

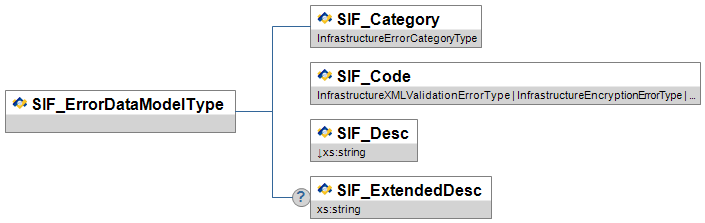
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Error.png)

Figure 5.1.9-1: SIF\_Error

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Error |  | This element is used to signal an unsuccessful response. |  |
|  | SIF\_Category | M |  | [InfrastructureErrorCategoryType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType) |
|  | SIF\_Code | M | See [Error Codes](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureErrorCategoryType). | union of:  [InfrastructureXMLValidationErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureXMLValidationErrorType) [InfrastructureEncryptionErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureEncryptionErrorType) [InfrastructureAuthenticationErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureAuthenticationErrorType) [InfrastructureAccessAndPermissionErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureAccessAndPermissionErrorType) [InfrastructureRegistrationErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureRegistrationErrorType) [InfrastructureProvisionErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureProvisionErrorType) [InfrastructureSubscriptionErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureSubscriptionErrorType) [InfrastructureRequestAndResponseErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureRequestAndResponseErrorType) [InfrastructureEventReportingAndProcessingErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureEventReportingAndProcessingErrorType) [InfrastructureTransportErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureTransportErrorType) [InfrastructureSystemErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureSystemErrorType) [InfrastructureGenericMessageHandlingErrorType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#InfrastructureGenericMessageHandlingErrorType) [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | 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." | [xs:string](http://www.w3.org/TR/xmlschema-2/#string)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 1024 | |
|  | 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)." | [xs:string](http://www.w3.org/TR/xmlschema-2/#string) |

Table 5.1.9-1: SIF\_Error

### 5.1.10 SIF\_Query

SIF's default query mechanism.

Note: With SIF Implementation Specification version 2.4 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.

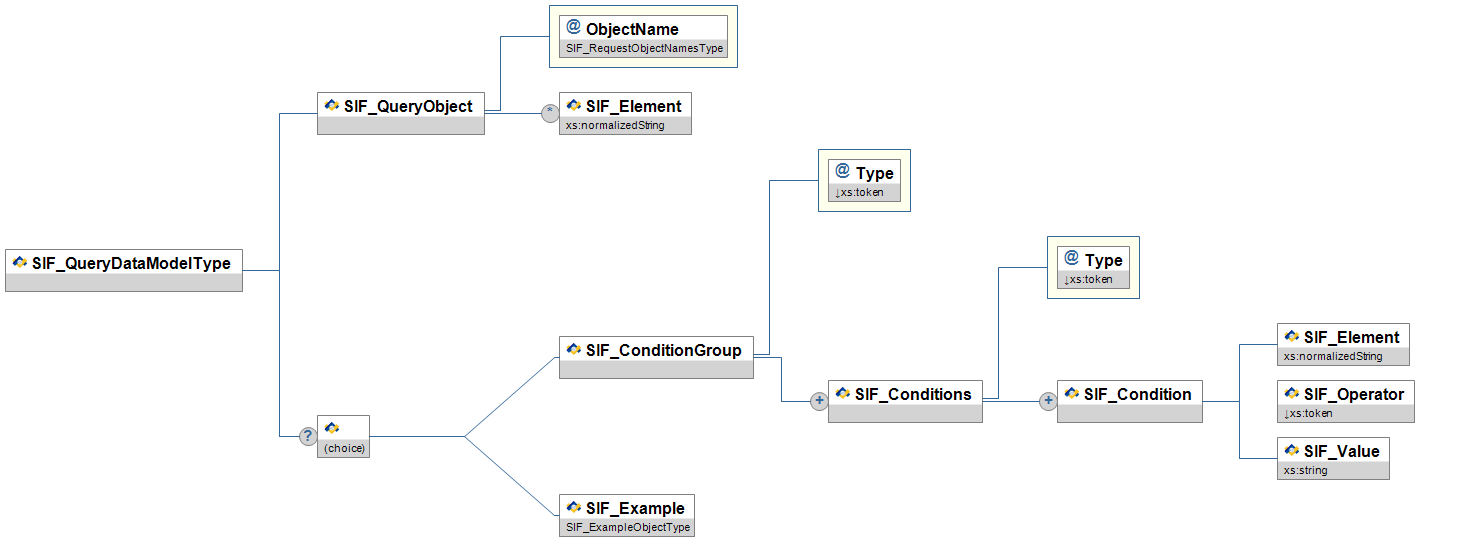
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Query.png)

Figure 5.1.10-1: SIF\_Query

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Query |  | SIF's default query mechanism.  Note: With SIF Implementation Specification version 2.4 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. |  |
|  | SIF\_QueryObject | M | This is the object that is being queried for. |  |
| **@** | ObjectName | M | The actual name of the object that is being queried for. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_QueryObject/SIF\_Element | OR | Individual elements/attributes being requested of matching object. See [SIF\_Element Syntax](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_ElementSyntax) 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.  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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
|  | SIF\_ConditionGroup | C | Either SIF\_ConditionGroup or SIF\_Example may optionally be specified to present conditions matching objects should satisfy.  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. |  |
| **@** | 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](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
|  | 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](http://www.w3.org/TR/xmlschema-2/#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. | [SIF\_ExampleObjectType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_ExampleObjectType) |

Table 5.1.10-1: SIF\_Query

#### 5.1.10.1 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 5.1.10.1-1

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 5.1.10.1-2

<SIF\_Message Version="2.4" 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 5.1.10.1-3: SIF\_ConditionGroup querying into an object

#### 5.1.10.2 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]](http://specification.sifinfo.org/Implementation/2.4/References.html" \l "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 withs 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 5.1.10.2-1

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 5.1.10.2-2

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

### 5.1.11 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 is 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.

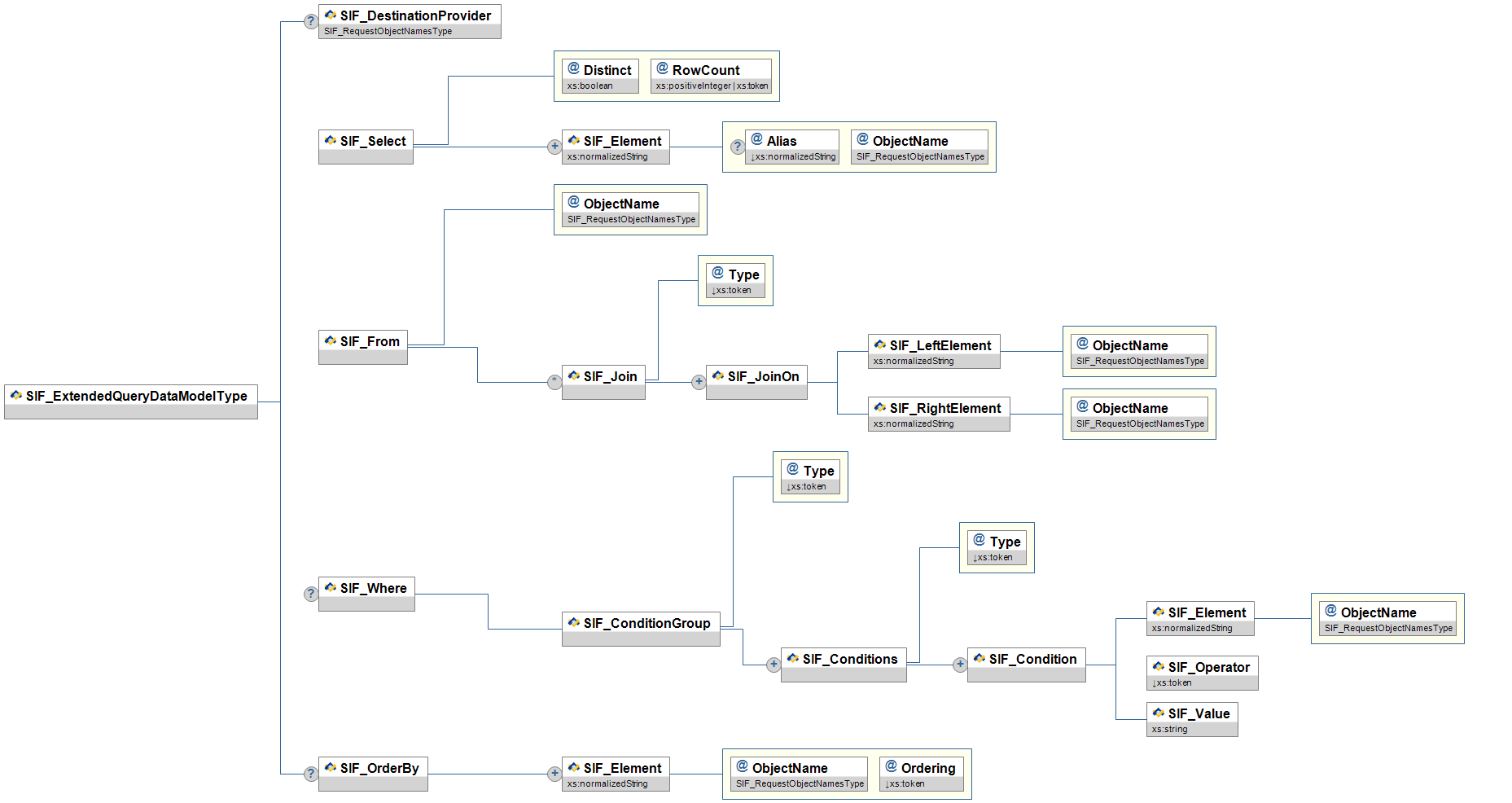
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_ExtendedQuery.png)

Figure 5.1.11-1: SIF\_ExtendedQuery

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | 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 is 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. |  |
|  | SIF\_DestinationProvider | O | If no SIF\_DestinationId 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 SIF\_DestinationId applies to the request, the ZIS routes the request to the Provider on record for the object name in SIF\_From. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_Select | M | Identifies which data elements/attribute are to be returned as columns in each matching row. |  |
| **@** | Distinct | M | Specifies whether query results should return all rows (false) or just distinct ones (true). Rows are distinct if at least one column differs between them. | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
| **@** | RowCount | M | The maximum number of rows to return. If All, return all rows, otherwise return the top rows up to the maximum row count indicated. | union of:  [xs:positiveInteger](http://www.w3.org/TR/xmlschema-2/#positiveInteger) additional values:  All |
|  | 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 ObjectName, or \* can be designated to return all attributes and immediate child elements of the object specified in ObjectName, or [SIF\_Element Syntax](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_ElementSyntax) can be specified, relative to the object specified in ObjectName. 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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
| **@** | Alias | O | Optional caption for the column. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
| **@** | ObjectName | M | The name of the object from which to retrieve element/attributes. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | 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 SIF\_Join clause. This clause generates the matching rows returned, optionally limited/filtered by the SIF\_Where clause. Each object referenced in the SIF\_Select, SIF\_Where and SIF\_OrderBy clauses must be included here. |  |
| **@** | ObjectName | M | The name of the object to query. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_From/SIF\_Join | OR | Additional objects to query, with join conditions specifying the relationships between objects. |  |
| **@** | Type | M | Type of relational join. | values:  Inner  LeftOuter  RightOuter  FullOuter |
|  | 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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
| **@** | ObjectName | M | Name of the object that contains the element/attribute. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | 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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
| **@** | ObjectName | M | Name of the object that contains the element/attribute. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_Where | O | Optionally specifies conditions to limit/filter rows resulting from the SIF\_From clause. |  |
|  | SIF\_Where/SIF\_ConditionGroup | M | Conditions that matching rows must meet. |  |
| **@** | 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\_Where/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\_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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
| **@** | ObjectName | M | The name of the object containing the element/attribute. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_Where/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\_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](http://www.w3.org/TR/xmlschema-2/#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](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
| **@** | ObjectName | M | The name of the object containing the element/attribute. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
| **@** | Ordering | M | Whether to order the element/attribute in ascending or descending order. | values:  Ascending  Descending |

Table 5.1.11-1: SIF\_ExtendedQuery

<SIF\_ExtendedQuery>

<SIF\_Select Distinct="false" RowCount="All">

<SIF\_Element ObjectName="StudentPersonal" />

</SIF\_Select>

<SIF\_From ObjectName="StudentPersonal" />

</SIF\_ExtendedQuery>

Example 5.1.11-1: 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 5.1.11-3: 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 5.1.11-5: 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 5.1.11-7: 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 5.1.11-9: Selecting a specific StudentPersonal's StudentSchoolEnrollment objects, along with the corresponding school name for each enrollment

#### 5.1.11.1 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 5.1.11.1-1: 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 5.1.11.1-1: 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 5.1.11.1-2: Corresponding SIF\_ExtendedQuery

### 5.1.12 SIF\_ExtendedQueryResults

This element provides a wrapper for data returned in response to a SIF\_ExtendedQuery. Used in SIF\_Response and SIF\_ReportObject.

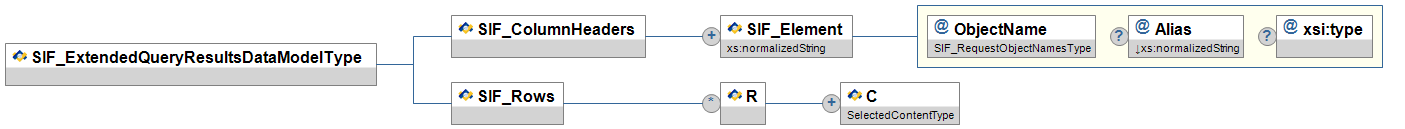
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_ExtendedQueryResults.png)

Figure 5.1.12-1: 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](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
| **@** | ObjectName | M | The object in which the element/attribute occurs. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
| **@** | Alias | O | The caption for the column, if specified in SIF\_ExtendedQuery. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
| **@** | 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](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SelectedContentType) |

Table 5.1.12-1: 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" SchoolInfoRefId="AED4AEF825284D7E9F082EBBEBB12345" MembershipType="Home" TimeFrame="Current" SchoolYear="2007">

<!--...-->

</StudentSchoolEnrollment>

</C>

<C>

<SchoolName>SIFA High</SchoolName>

</C>

</R>

<R>

<C>

<StudentPersonal RefId="12345678901234567890123456789012">

<!--...-->

</StudentPersonal>

</C>

<C>

<StudentSchoolEnrollment RefId="AED4AEF825284D7E9F082EBBEB1999FA" StudentPersonalRefId="12345678901234567890123456789012" SchoolInfoRefId="ED4AEF825284D7E9F082EBBEBB678902" MembershipType="Concurrent" TimeFrame="Current" SchoolYear="2007">

<!--...-->

</StudentSchoolEnrollment>

</C>

<C>

<SchoolName>SIFA University</SchoolName>

</C>

</R>

</SIF\_Rows>

</SIF\_ExtendedQueryResults>

Example 5.1.12-1: 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.

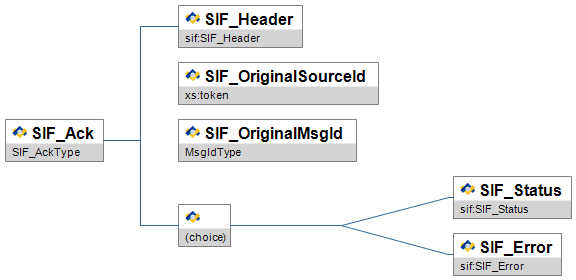
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Ack.png)

Figure 5.2.1-1: SIF\_Ack

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Ack | M | This message is used as an acknowledgement to an infrastructure message. |  |
|  | SIF\_Header | M | Header information associated with this message. | [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_OriginalSourceId | M | The SIF\_SourceId of the infrastructure message for which the SIF\_Ack serves as a response. | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_OriginalMsgId | M | The SIF\_MsgId of the infrastructure message for which the SIF\_Ack message serves as a response. | [MsgIdType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#MsgIdType) |
|  | SIF\_Status | C | This element is used to signal a successful response. | [SIF\_Status](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Status) |
|  | SIF\_Error | C | This element is used to signal an unsuccessful response. | [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) |

Table 5.2.1-1: SIF\_Ack

<SIF\_Message Version="2.4" 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="2.4">

<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="2.4" 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 5.2.1-3: 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.

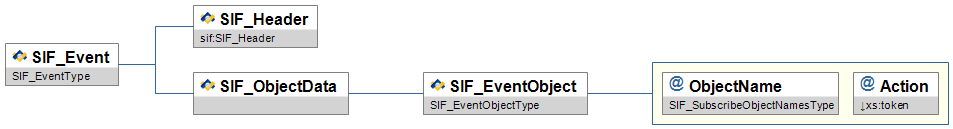
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Event.png)

Figure 5.2.2-1: SIF\_Event

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Event | M | SIF\_Event is used to deliver event objects as defined in SIF. |  |
|  | SIF\_Header | M | Header information associated with this message. | [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_ObjectData | M |  |  |
|  | SIF\_ObjectData/SIF\_EventObject | M | Contains the actual object (partial or whole) that is being added, changed or deleted. | [SIF\_EventObjectType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_EventObjectType) |
| **@** | ObjectName | M | This is the name of the object being added, changed or deleted. | [SIF\_SubscribeObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_SubscribeObjectNamesType) |
| **@** | Action | M | This is the action associated with the object that is being conveyed by this SIF\_Event. | values:  Add  Delete  Change |

Table 5.2.2-1: SIF\_Event

<SIF\_Message Version="2.4" 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.

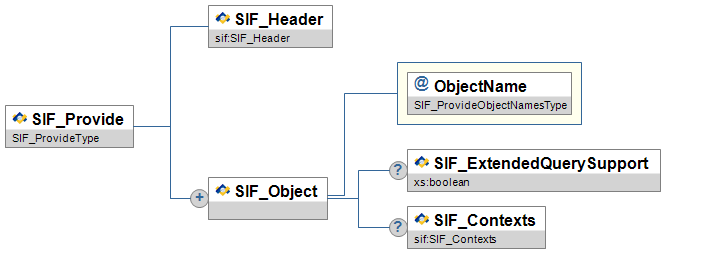
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Provide.png)

Figure 5.2.3-1: SIF\_Provide

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | SIF\_Provide | M | The SIF\_Provide message is used for advertising the provision of data objects. |  |
|  | SIF\_Header | M | Header information associated with this message. | [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Object | MR | This is the object that is being provided. |  |
| **@** | ObjectName | M | The actual name of the object that is being provided. | [SIF\_ProvideObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_ProvideObjectNamesType) |
|  | SIF\_Object/SIF\_ExtendedQuerySupport | O | Whether or not the Agent supports SIF\_ExtendedQuery for this object. | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
|  | SIF\_Object/SIF\_Contexts | O | The contexts in which the object is being provided; if omitted, the context is SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |

Table 5.2.3-1: SIF\_Provide

<SIF\_Message Version="2.4" 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.

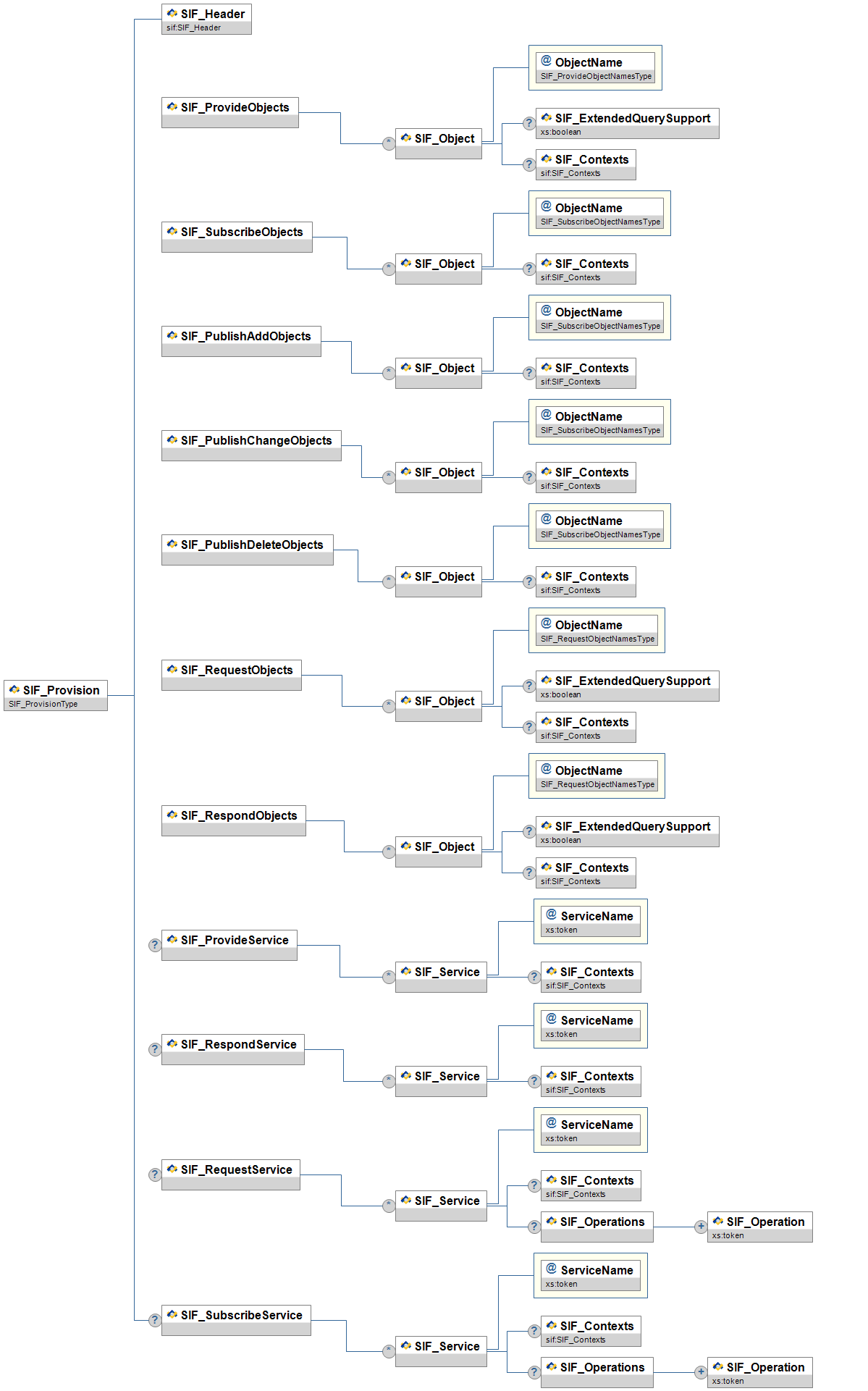
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Provision.png)

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 SIF\_Provision, some operations may still be rejected with access control errors. |  |
|  | SIF\_Header | M | Header information associated with this message. | [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_ProvideObjects | M | A list of objects an Agent wishes to provide. |  |
|  | SIF\_ProvideObjects/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [SIF\_ProvideObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_ProvideObjectNamesType) |
|  | SIF\_ProvideObjects/SIF\_Object/      SIF\_ExtendedQuerySupport | O | Whether or not SIF\_ExtendedQuery is supported with regard to this object. | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
|  | SIF\_ProvideObjects/SIF\_Object/      SIF\_Contexts | O | Applicable contexts for stated object support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | 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. | [SIF\_SubscribeObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_SubscribeObjectNamesType) |
|  | SIF\_SubscribeObjects/SIF\_Object/      SIF\_Contexts | O | Applicable contexts for stated object support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_PublishAddObjects | M | A list of objects for which an Agent wishes to publish Add events. |  |
|  | SIF\_PublishAddObjects/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [SIF\_SubscribeObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_SubscribeObjectNamesType) |
|  | SIF\_PublishAddObjects/SIF\_Object/      SIF\_Contexts | O | Applicable contexts for stated object support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_PublishChangeObjects | M | A list of objects for which an Agent wishes to publish Change events. |  |
|  | SIF\_PublishChangeObjects/      SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [SIF\_SubscribeObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_SubscribeObjectNamesType) |
|  | SIF\_PublishChangeObjects/      SIF\_Object/SIF\_Contexts | O | Applicable contexts for stated object support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_PublishDeleteObjects | M | A list of objects for which an Agent wishes to publish Delete events. |  |
|  | SIF\_PublishDeleteObjects/      SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [SIF\_SubscribeObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_SubscribeObjectNamesType) |
|  | SIF\_PublishDeleteObjects/      SIF\_Object/SIF\_Contexts | O | Applicable contexts for stated object support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_RequestObjects/SIF\_Object/      SIF\_ExtendedQuerySupport | O | Optionally specify whether or not SIF\_ExtendedQuery may be sent in requests for this object. | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
|  | SIF\_RequestObjects/SIF\_Object/      SIF\_Contexts | O | Applicable contexts for stated object support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | 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. Note that this list MUST include those objects specified in SIF\_ProvideObjects. |  |
|  | SIF\_RespondObjects/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_RespondObjects/SIF\_Object/      SIF\_ExtendedQuerySupport | O | Whether or not SIF\_ExtendedQuery is supported with regard to this object. | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
|  | SIF\_RespondObjects/SIF\_Object/      SIF\_Contexts | O | Applicable contexts for stated object support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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 | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ProvideService/SIF\_Service/      SIF\_Contexts | O | Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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 | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_RespondService/SIF\_Service/      SIF\_Contexts | O | Applicable contexts for stated SIF Zone Service support. If omitted, the context defaults to SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | 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 optinal 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 | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_RequestService/SIF\_Service/      SIF\_Operations/SIF\_Operation | MR | A specific method that the agent invokes the specified SIF Zone Service | [xs:token](http://www.w3.org/TR/xmlschema-2/#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 | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_SubscribeService/SIF\_Service/      SIF\_Operations/SIF\_Operation | MR | A specific event that the agent desires to subscribe to | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |

Table 5.2.4-1: SIF\_Provision

<SIF\_Message Version="2.4" 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.

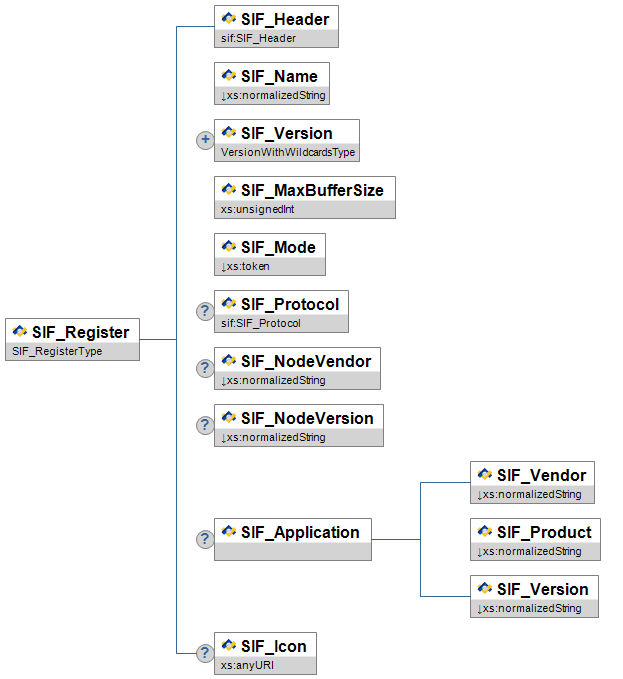
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Register.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Name | M | This is the descriptive name of the agent that is registering (i.e. Ramsey Media Center). | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_Version | MR | 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.  The format of SIF\_Version values can be found in [Version Numbers](http://specification.sifinfo.org/Implementation/2.4/Introduction.html#VersionNumbers). In a SIF\_Register message, an individual SIF\_Version element may also contain the following wildcards:  \* - Any SIF version  <major version>.\* - Any minor version plus revisions within a major version (e.g., 1.\*)  <major version>.<minor version><r>\* - Any revision within a minor version (e.g., 1.1r\*)  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. | [VersionWithWildcardsType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#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](http://www.w3.org/TR/xmlschema-2/#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Protocol) |
|  | SIF\_NodeVendor | O | The vendor of the SIF agent. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |
|  | 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 | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 32 | |
|  | 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](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |
|  | SIF\_Application/SIF\_Product | M | The name of the product that this agent supports. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |
|  | SIF\_Application/SIF\_Version | M | The version of the product. This field is informative only. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 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 image MIME type commonly supported by Web browsers (e.g. PNG, JPEG, GIF). Agents may optionally follow the more restrictive UUIDelines at [[FAVICON]](http://specification.sifinfo.org/Implementation/2.4/References.html#FAVICON). | [xs:anyURI](http://www.w3.org/TR/xmlschema-2/#anyURI) |

Table 5.2.5-1: SIF\_Register

<SIF\_Message Version="2.4" 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>2.4</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.

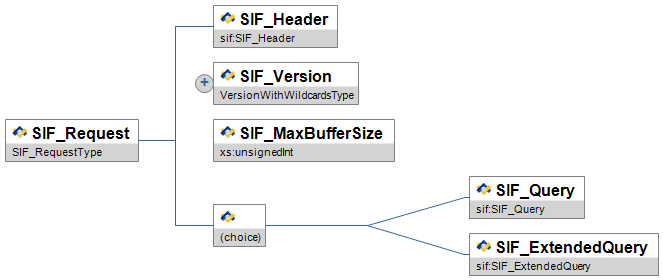
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Request.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#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](http://www.w3.org/TR/xmlschema-2/#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Query) |
|  | SIF\_ExtendedQuery | C |  | [SIF\_ExtendedQuery](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_ExtendedQuery) |

Table 5.2.6-1: SIF\_Request

<SIF\_Message Version="2.4" 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.

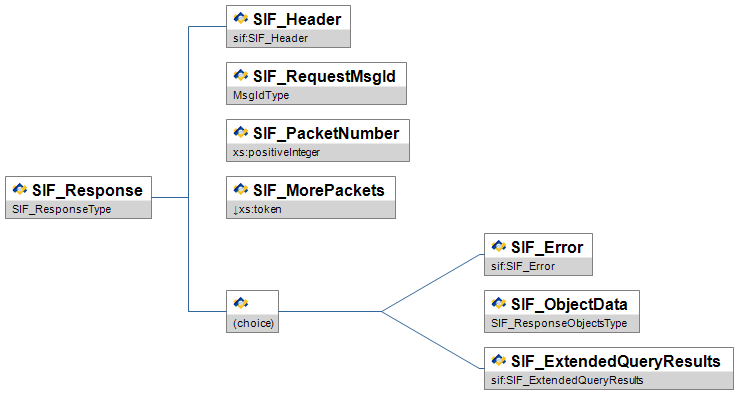
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Response.png)

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. |  |
|  | SIF\_Header | M | Header information associated with this message. The SIF\_DestinationId needs to be the SIF\_SourceId of the original SIF\_Request message being processed. | [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_RequestMsgId | M | This is the message Id of the SIF\_Request message being processed. It provides a unique match between a SIF\_Response and a previous SIF\_Request. Since the Id of each message from an agent is unique, the receiver of a SIF\_Response message will be able to relate the SIF\_Response to a SIF\_Request that it sent out previously. | [MsgIdType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#MsgIdType) |
|  | SIF\_PacketNumber | M | This element represents the index of the SIF\_Response 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.  The receiver of a SIF\_Response message, with the help of the SIF\_MorePackets and SIF\_PacketNumber element in each incoming SIF\_Response message, will be able to interpret and process each SIF\_Response as part of a complete response to a previous SIF\_Request. | [xs:positiveInteger](http://www.w3.org/TR/xmlschema-2/#positiveInteger) |
|  | SIF\_MorePackets | M | This element provides an indication as to whether there are more packets besides this one to make up a complete response.  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\_Request. Agents may also break response data into multiple packets for the benefit of improving performance or for circumventing limitations of the underlying network infrastructure.  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\_Request as indicated by the SIF\_RequestMsgId element. | values:  Yes  No |
|  | SIF\_Error | C | The responder returns SIF\_Error, SIF\_ObjectData or SIF\_ExtendedQueryResults.  This element allows the Responder to report an error condition that occurs while processing the SIF\_Request.  If a SIF\_Error element is present, the requesting agent must not expect to receive further SIF\_Responses to the SIF\_Request. | [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) |
|  | SIF\_ObjectData | C | The SIF\_ObjectData element contains the data objects matching the supplied criteria in the SIF\_Request message if the SIF\_Request contained SIF\_Query. If the SIF\_Request contained SIF\_ExtendedQuery, include SIF\_ExtendedQueryResults. | [SIF\_ResponseObjectsType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_ResponseObjectsType) |
|  | SIF\_ExtendedQueryResults | C | This element contains the elements requested by SIF\_ExtendedQuery in SIF\_Request. | [SIF\_ExtendedQueryResults](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_ExtendedQueryResults) |

Table 5.2.7-1: SIF\_Response

<SIF\_Message Version="2.4" 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="2.4" 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="2.4" 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="2.4" 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.

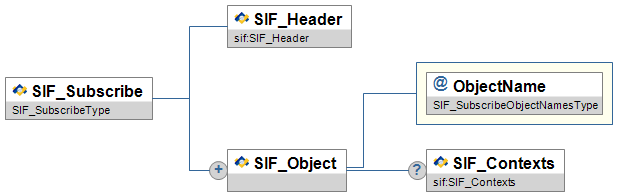
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Subscribe.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The actual name of the object that is being subscribed to. All valid SIF\_Events for this object will be routed to the subscriber. | [SIF\_SubscribeObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_SubscribeObjectNamesType) |
|  | SIF\_Object/SIF\_Contexts | O | The contexts to which the subscription applies; if omitted, the context is SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |

Table 5.2.8-1: SIF\_Subscribe

<SIF\_Message Version="2.4" 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.

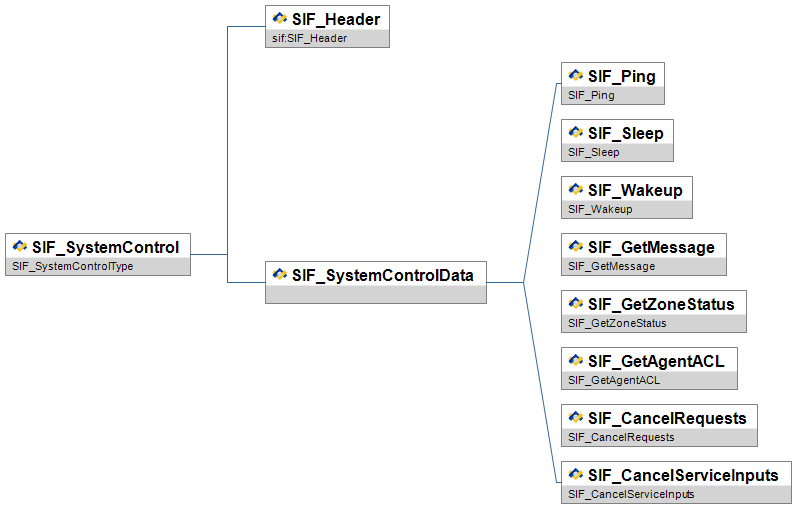
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_SystemControl.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_SystemControlData | M | This element holds the sub-message being sent. | choice of:  [SIF\_Ping](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Ping) [SIF\_Sleep](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Sleep) [SIF\_Wakeup](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Wakeup) [SIF\_GetMessage](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_GetMessage) [SIF\_GetZoneStatus](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_GetZoneStatus) [SIF\_GetAgentACL](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_GetAgentACL) [SIF\_CancelRequests](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_CancelRequests) [SIF\_CancelServiceInputs](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_CancelServiceInputs) |

Table 5.2.9-1: SIF\_SystemControl

<SIF\_Message Version="2.4" 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.

[SIF_Ping](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Ping.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#EMPTY) |

Table 5.2.10-1: SIF\_Ping

<SIF\_Message Version="2.4" 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="2.4" 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="2.4" 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="2.4" 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.

[SIF_Sleep](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Sleep.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#EMPTY) |

Table 5.2.11-1: SIF\_Sleep

<SIF\_Message Version="2.4" 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="2.4" 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.

[SIF_Wakeup](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Wakeup.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#EMPTY) |

Table 5.2.12-1: SIF\_Wakeup

<SIF\_Message Version="2.4" 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="2.4" 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.

[SIF_GetMessage](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_GetMessage.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#EMPTY) |

Table 5.2.13-1: SIF\_GetMessage

<SIF\_Message Version="2.4" 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="2.4" 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="2.4">

<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="2.4" 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.

[SIF_GetZoneStatus](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_GetZoneStatus.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#EMPTY) |

Table 5.2.14-1: SIF\_GetZoneStatus

<SIF\_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">

<SIF\_SystemControl>

<SIF\_Header>

<SIF\_MsgId>91401B5073F54AB1AEBC63E51764C77A</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="2.4" 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.

[SIF_GetAgentACL](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_GetAgentACL.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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."

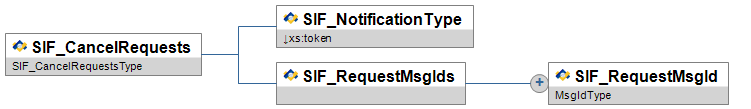
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_CancelRequests.png)

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 SIF\_Requests, pending or in process. |  |
|  | SIF\_NotificationType | M |  | values:  Standard  ZIS will send a "final" SIF\_Response for each cancelled SIF\_Request.  None  No further SIF\_Responses for these requests will be placed in the Agent's queue. |
|  | SIF\_RequestMsgIds | M | The list of SIF\_Requests to cancel. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_RequestMsgIds/SIF\_RequestMsgId | MR | This is the SIF\_MsgId of the SIF\_Request message being cancelled. | [MsgIdType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#MsgIdType) |

Table 5.2.16-1: SIF\_CancelRequests

<SIF\_Message Version="2.4" 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.

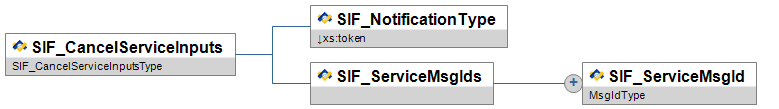
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_CancelServiceInputs.png)

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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ServiceMsgIds/SIF\_ServiceMsgId | MR | This is the SIF\_ServiceMsgId of the SIF\_ServiceInput message being cancelled. | [MsgIdType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#MsgIdType) |

Table 5.2.17-1: SIF\_CancelServiceInputs

<SIF\_Message Version="2.4" 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.

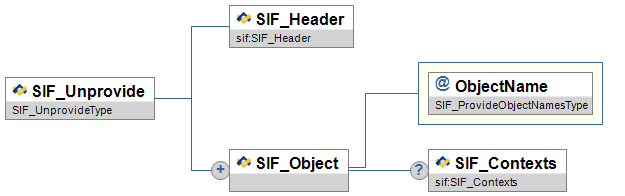
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Unprovide.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Object | MR | This is the object that is being removed from the provider list. |  |
| **@** | ObjectName | M | The actual name of the object that is being removed. | [SIF\_ProvideObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_ProvideObjectNamesType) |
|  | SIF\_Object/SIF\_Contexts | O | The contexts from which the object is being removed; if omitted, the context is SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |

Table 5.2.18-1: SIF\_Unprovide

<SIF\_Message Version="2.4" 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.

[SIF_Unregister](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Unregister.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |

Table 5.2.19-1: SIF\_Unregister

<SIF\_Message Version="2.4" 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.

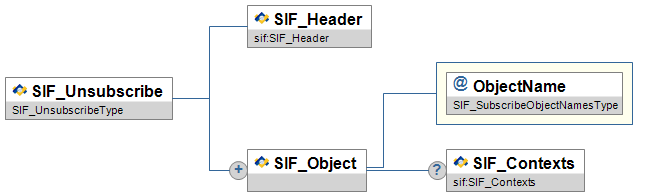
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_Unsubscribe.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The actual name of the 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. | [SIF\_SubscribeObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_SubscribeObjectNamesType) |
|  | SIF\_Object/SIF\_Contexts | O | The applicable contexts; if omitted, the context is SIF\_Default. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |

Table 5.2.20-1: SIF\_Unsubscribe

<SIF\_Message Version="2.4" 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.

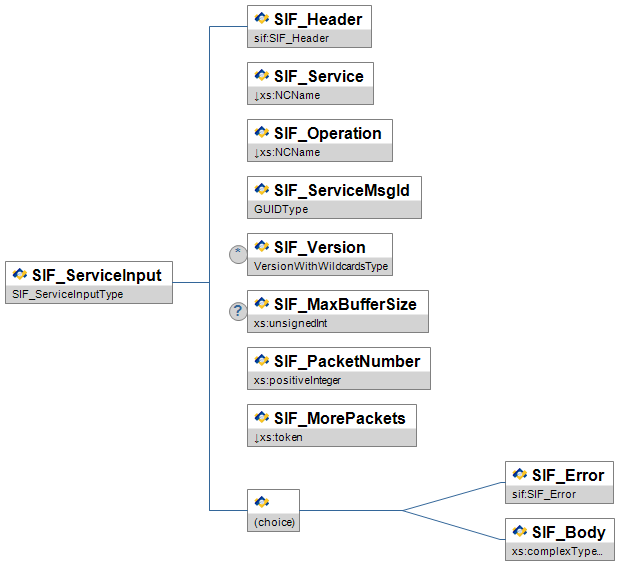
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_ServiceInput.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Service | M | The name of the SIF Zone Service that is being invoked. | values:  serviceAgencyStudentIdRequest  serviceAgencyStudentIdManagement  serviceAgencyStaffIdRequest  serviceAgencyStaffIdManagement  serviceAssessmentProcessing  serviceAssessmentAdministration  serviceSREBroker  serviceSREPublisher  serviceSREAcademicRecordConsumer  serviceSREDemographicRecordConsumer  serviceSRESpecialEducationRecordConsumer |
|  | SIF\_Operation | M | This is the name of the service operation that is being invoked. | values:  RequestIdentifier  AssignIdentifier  ResolveIdentifier  ReleaseIdentifier  CancelTransaction  ScoreItems  ItemsScored  GetRecordExchange  SetRecordExchange  SetAcademicRecord  SetDemographicRecord  SetSpecialEducationRecord |
|  | SIF\_ServiceMsgId | M | The unique Id of this service request invocation. Multiple SIF\_ServiceInput packets may be sent with this same SIF\_ServiceMsgId | [UUIDType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html" \l "GUIDType) |
|  | SIF\_Version | CR | 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.\*".  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. | [VersionWithWildcardsType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#VersionWithWildcardsType) |
|  | SIF\_MaxBufferSize | C | 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.  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. | [xs:unsignedInt](http://www.w3.org/TR/xmlschema-2/#unsignedInt) |
|  | SIF\_PacketNumber | M | 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.  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. | [xs:positiveInteger](http://www.w3.org/TR/xmlschema-2/#positiveInteger) |
|  | SIF\_MorePackets | M | 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".  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.  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. | values:  Yes  No |
|  | SIF\_Error | C | 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.  If a SIF\_Error element is present, the service must not expect to receive further SIF\_ServiceInput messagest. | [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) |
|  | SIF\_Body | C | 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. | <xs:complexType>  <xs:sequence>  <xs:any processContents="lax" />  </xs:sequence>  </xs:complexType> |

Table 5.2.21-1: SIF\_ServiceInput

<SIF\_Message Version="2.4" 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.

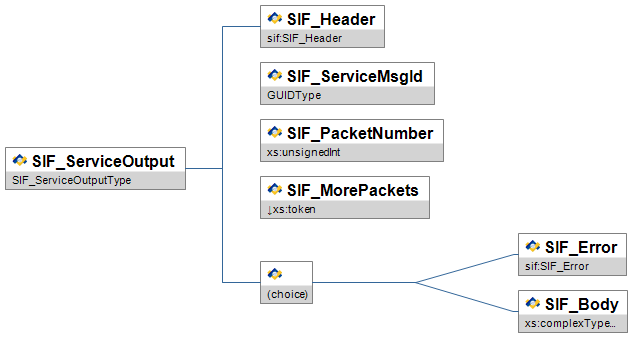
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_ServiceOutput.png)

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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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. | [UUIDType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html" \l "GUIDType) |
|  | SIF\_PacketNumber | M | 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.  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. | [xs:unsignedInt](http://www.w3.org/TR/xmlschema-2/#unsignedInt) |
|  | SIF\_MorePackets | M | 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".  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.  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. | values:  Yes  No |
|  | SIF\_Error | C | This element allows the Responder to report an error condition that occurs while processing the SIF\_ServiceInput.  If a SIF\_Error element is present, the requesting agent must not expect to receive further SIF\_ServiceOutputs to the SIF\_ServiceInput. | [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) |
|  | SIF\_Body | C | 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. | <xs:complexType>  <xs:sequence>  <xs:any processContents="lax" />  </xs:sequence>  </xs:complexType> |

Table 5.2.22-1: SIF\_ServiceOutput

<SIF\_Message Version="2.4" 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.

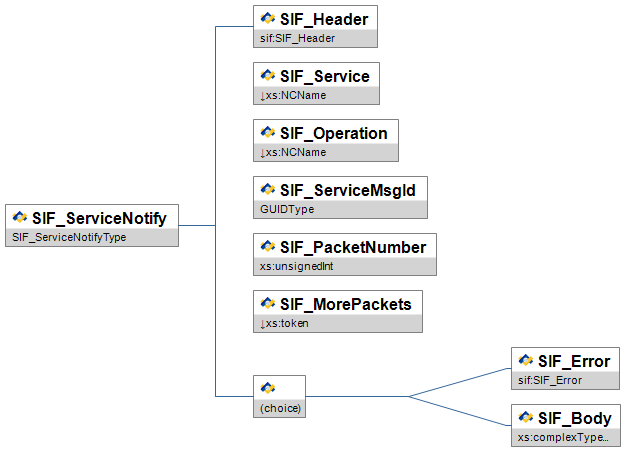
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_ServiceNotify.png)

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. |  |
|  | SIF\_Header | M | Header information associated with this message. | [SIF\_Header](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Service | M | The name of the SIF Zone Service that initiated the event | values:  serviceAssessmentAdministration |
|  | SIF\_Operation | M | The name of the notification message being sent | values:  AdministrationScored |
|  | SIF\_ServiceMsgId | M | A UUID that has been assigned to this series of messages. All SIF\_ServiceNotify messages that contain this Id correspond to the same event instance. | [UUIDType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html" \l "GUIDType) |
|  | SIF\_PacketNumber | M | 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.  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. | [xs:unsignedInt](http://www.w3.org/TR/xmlschema-2/#unsignedInt) |
|  | SIF\_MorePackets | M | 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".  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.  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. | values:  Yes  No |
|  | SIF\_Error | C | 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.  If a SIF\_Error element is present, the receiver must not expect to receive further SIF\_ServiceNotify messages as part of this message stream. | [SIF\_Error](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Error) |
|  | SIF\_Body | C | 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. | <xs:complexType>  <xs:sequence>  <xs:any processContents="lax" />  </xs:sequence>  </xs:complexType> |

Table 5.2.23-1: SIF\_ServiceNotify

<SIF\_Message Version="2.4" 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.

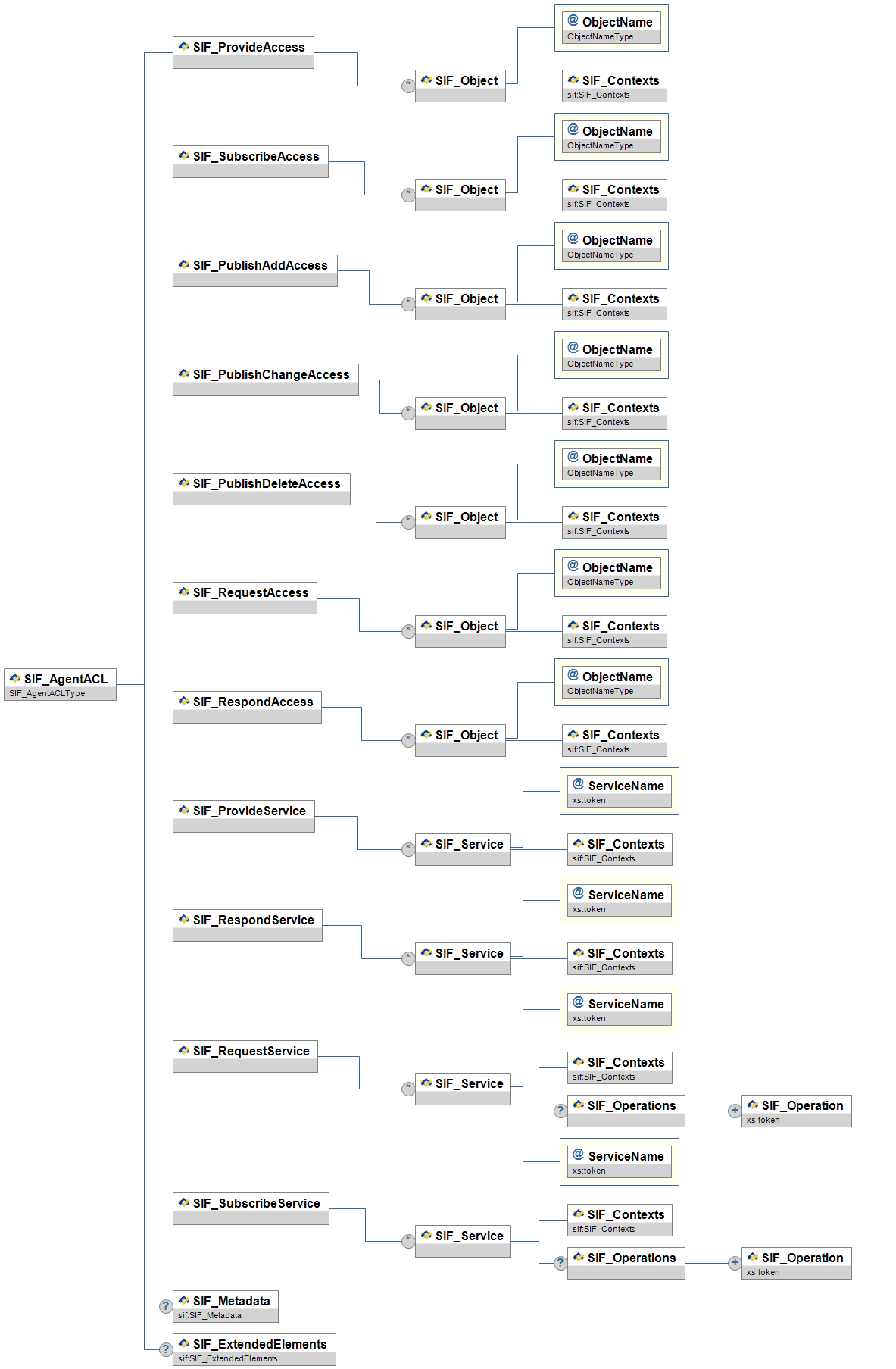
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_AgentACL.png)

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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ProvideAccess/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_ProvideAccess/SIF\_Object/      SIF\_Contexts | M | List of contexts in which rights for the given action/object apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_SubscribeAccess | M | Access control list by object for SIF\_Subscribe and corresponding section in SIF\_Provision. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_SubscribeAccess/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_SubscribeAccess/SIF\_Object/      SIF\_Contexts | M | List of contexts in which rights for the given action/object apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_PublishAddAccess/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_PublishAddAccess/SIF\_Object/      SIF\_Contexts | M | List of contexts in which rights for the given action/object apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_PublishChangeAccess/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_PublishChangeAccess/SIF\_Object/      SIF\_Contexts | M | List of contexts in which rights for the given action/object apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_PublishDeleteAccess/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_PublishDeleteAccess/SIF\_Object/      SIF\_Contexts | M | List of contexts in which rights for the given action/object apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_RequestAccess | M | Access control list by object for SIF\_Request and corresponding section in SIF\_Provision. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_RequestAccess/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_RequestAccess/SIF\_Object/      SIF\_Contexts | M | List of contexts in which rights for the given action/object apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_RespondAccess/SIF\_Object | OR |  |  |
| **@** | ObjectName | M | The name of each object. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_RespondAccess/SIF\_Object/      SIF\_Contexts | M | List of contexts in which rights for the given action/object apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_ProvideService | M | Indicates that the recipient agent has permission to provide one or more services to the SIF Zone | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ProvideService/SIF\_Service/      SIF\_Contexts | M | List of contexts in which rights for the given action/service apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_RespondService/SIF\_Service | OR |  |  |
| **@** | ServiceName | M | The name of the SIF Zone Service as defined by a SIF Zone Service specification | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_RespondService/SIF\_Service/      SIF\_Contexts | M | List of contexts in which rights for the given action/service apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_RequestService | M | Indicates that the recipient agent has permission to make service calls to a SIF Zone Service | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_RequestService/SIF\_Service | OR |  |  |
| **@** | ServiceName | M | The name of the SIF Zone Service as defined by a SIF Zone Service specification | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_RequestService/SIF\_Service/      SIF\_Contexts | M | List of contexts in which rights for the given action/service apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | 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. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_RequestService/SIF\_Service/      SIF\_Operations/SIF\_Operation | MR | A specific operation that the agent has permission to invoke or subscribe to | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_SubscribeService | M | Indicates that the recipient agent has permission to subscribe to notification messages that are emitted from a SIF Service | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_SubscribeService/SIF\_Service | OR |  |  |
| **@** | ServiceName | M | The name of the SIF Zone Service as defined by a SIF Zone Service specification | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_SubscribeService/SIF\_Service/      SIF\_Contexts | M | List of contexts in which rights for the given action/service apply. | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_SubscribeService/SIF\_Service/      SIF\_Operations | O | If SIF\_Operations is not present, then the agent has permission to subcribe to all notifications in the specified service . | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_Metadata | O |  | [SIF\_Metadata](http://specification.sifinfo.org/Implementation/2.4/DataModelTaskForce.html#SIF_Metadata) |
|  | SIF\_ExtendedElements | O |  | [SIF\_ExtendedElements](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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.

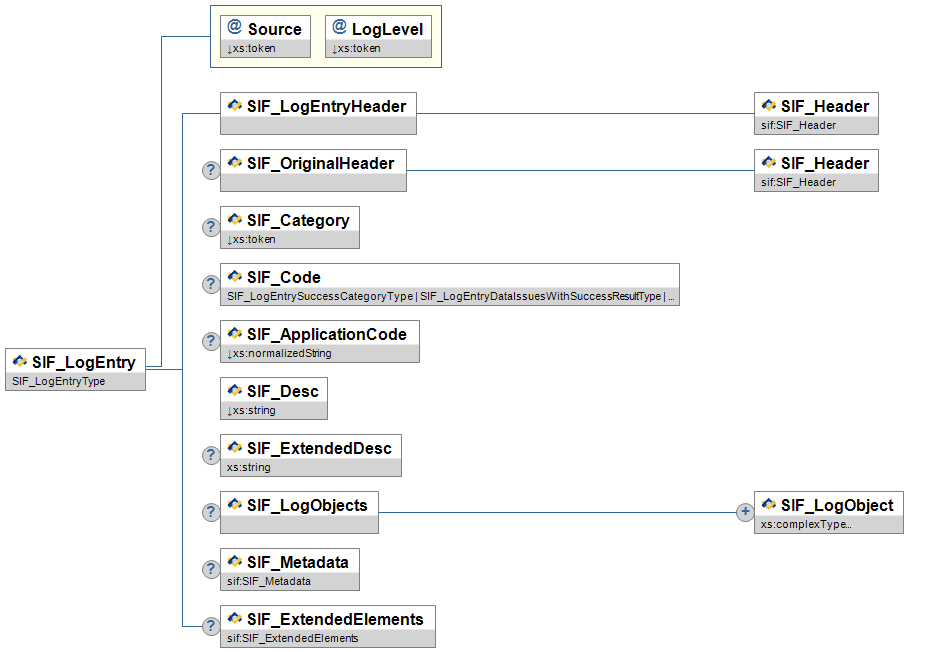
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_LogEntry.png)

Figure 5.3.2-1: SIF\_LogEntry

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | 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. |  |
| **@** | Source | M | The SIF node that logged this entry. | values:  Agent  ZIS |
| **@** | LogLevel | M | The level of the log entry herein described. | values:  Info  Warning  Error |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Header) |
|  | SIF\_Category | C | A SIF\_LogEntry category. May be omitted for informational-type postings, where typically a textual description will suffice.  Note that categories may be combined with the Source attribute of SIF\_LogEntry to differentiate agent error conditions from ZIS error conditions. | values:  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:  [SIF\_LogEntrySuccessCategoryType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#SIF_LogEntrySuccessCategoryType) [SIF\_LogEntryDataIssuesWithSuccessResultType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#SIF_LogEntryDataIssuesWithSuccessResultType) [SIF\_LogEntryDataIssuesWithFailureResultType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#SIF_LogEntryDataIssuesWithFailureResultType) [SIF\_LogEntryAgentErrorConditionType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#SIF_LogEntryAgentErrorConditionType) [SIF\_LogEntryZISErrorConditionType](http://specification.sifinfo.org/Implementation/2.4/CodeSets.html#SIF_LogEntryZISErrorConditionType) |
|  | 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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_Desc | M | A textual description of the error. | [xs:string](http://www.w3.org/TR/xmlschema-2/#string)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 1024 | |
|  | SIF\_ExtendedDesc | O | Any extended error description. | [xs:string](http://www.w3.org/TR/xmlschema-2/#string) |
|  | SIF\_LogObjects | O |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_LogObjects/SIF\_LogObject | MR | Any SIF data objects to which this log entry may apply. | <xs:complexType>  <xs:sequence>  <xs:any processContents="skip" />  </xs:sequence>  <xs:attribute name="ObjectName" use="required" type="sif:SIF\_RequestObjectNamesType" />  </xs:complexType> |
| **@** | ObjectName | M | The name of the SIF object referenced (e.g. StudentPersonal). | [SIF\_RequestObjectNamesType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#SIF_RequestObjectNamesType) |
|  | SIF\_Metadata | O |  | [SIF\_Metadata](http://specification.sifinfo.org/Implementation/2.4/DataModelTaskForce.html#SIF_Metadata) |
|  | SIF\_ExtendedElements | O |  | [SIF\_ExtendedElements](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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 of 16,384 bytes.</SIF\_Desc>

</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.

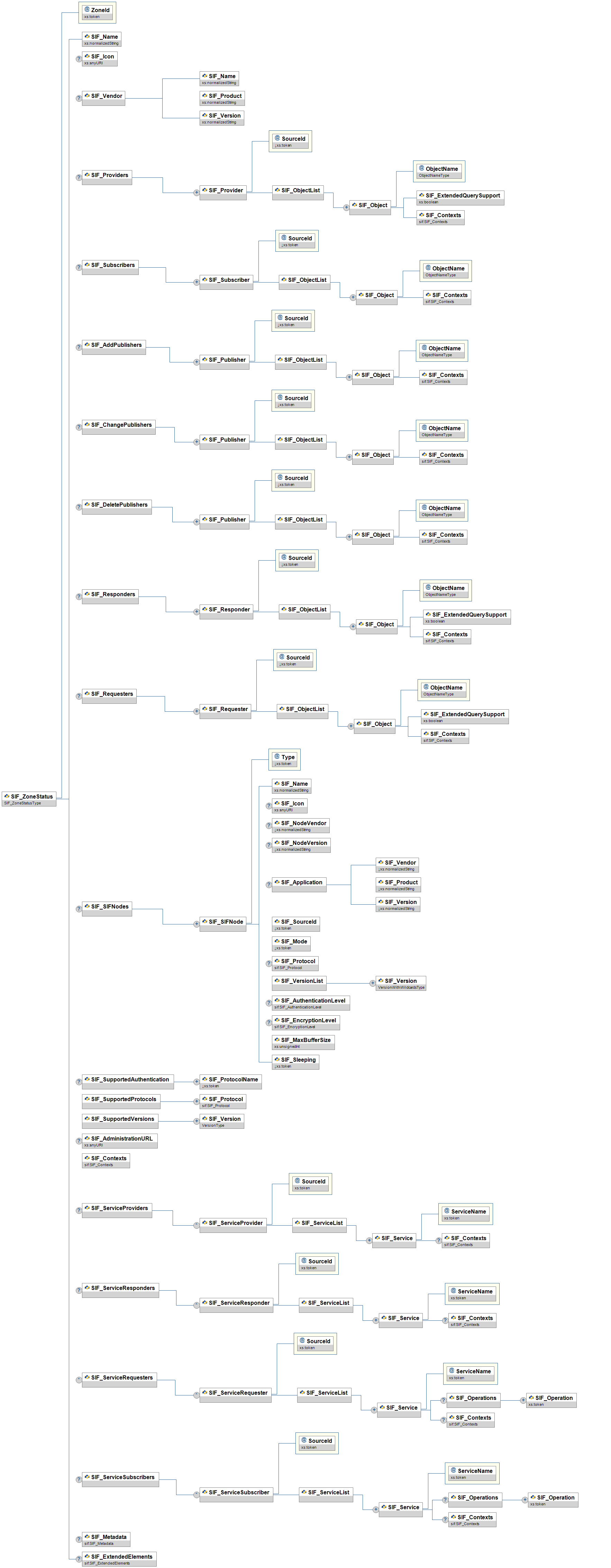
[](http://specification.sifinfo.org/Implementation/2.4/diagrams/SIF_ZoneStatus.png)

Figure 5.3.3-1: SIF\_ZoneStatus

|  | **Element/@Attribute** | **Char** | **Description** | **Type** |
| --- | --- | --- | --- | --- |
|  | 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. |  |
| **@ key** | ZoneId | M | The identifier for this Zone. It is the same as the SIF\_SourceId that the ZIS would place in any SIF\_Header that it creates. | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_Name | M | The descriptive name for the zone. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
|  | 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 image MIME type commonly supported by Web browsers (e.g. PNG, JPEG, GIF). Agents may optionally follow the more restrictive UUIDelines at [[FAVICON]](http://specification.sifinfo.org/Implementation/2.4/References.html#FAVICON). | [xs:anyURI](http://www.w3.org/TR/xmlschema-2/#anyURI) |
|  | 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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
|  | SIF\_Vendor/SIF\_Product | M | The product name assigned by the vendor to identify this ZIS. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
|  | SIF\_Vendor/SIF\_Version | M | The version of the vendor's product—not necessarily the SIF version. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
|  | SIF\_Providers | C | Encompasses all the providers registered with this ZIS. This element is mandatory if there are providers registered with the ZIS. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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 SIF\_SourceId field of any SIF\_Header created by the SIF node. | [xs:token](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_Providers/SIF\_Provider/      SIF\_ObjectList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_Providers/SIF\_Provider/      SIF\_ObjectList/SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The name of the object being provided by this SIF node. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_Providers/SIF\_Provider/      SIF\_ObjectList/SIF\_Object/      SIF\_ExtendedQuerySupport | M |  | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
|  | SIF\_Providers/SIF\_Provider/      SIF\_ObjectList/SIF\_Object/      SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_Subscribers/SIF\_Subscriber | MR |  |  |
| **@** | 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 SIF\_SourceId field of any SIF\_Header created by the SIF node. | [xs:token](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_Subscribers/SIF\_Subscriber/      SIF\_ObjectList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_Subscribers/SIF\_Subscriber/      SIF\_ObjectList/SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The name of the object being subscribed to by this SIF node. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_Subscribers/SIF\_Subscriber/      SIF\_ObjectList/SIF\_Object/      SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_AddPublishers | C | Encompasses all the Add SIF\_Event publishers registered with this zone. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_AddPublishers/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](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_AddPublishers/SIF\_Publisher/      SIF\_ObjectList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_AddPublishers/SIF\_Publisher/      SIF\_ObjectList/SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The name of the object being published by this agent. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_AddPublishers/SIF\_Publisher/      SIF\_ObjectList/SIF\_Object/      SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_ChangePublishers | C | Encompasses all the Change SIF\_Event publishers registered with this zone. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_ChangePublishers/SIF\_Publisher/      SIF\_ObjectList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ChangePublishers/SIF\_Publisher/      SIF\_ObjectList/SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The name of the object being published by this agent. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_ChangePublishers/SIF\_Publisher/      SIF\_ObjectList/SIF\_Object/      SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_DeletePublishers | C | Encompasses all the Delete SIF\_Event publishers registered with this zone. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_DeletePublishers/SIF\_Publisher/      SIF\_ObjectList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_DeletePublishers/SIF\_Publisher/      SIF\_ObjectList/SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The name of the object being published by this agent. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_DeletePublishers/SIF\_Publisher/      SIF\_ObjectList/SIF\_Object/      SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_Responders | C | Encompasses all the responders registered with this zone. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_Responders/SIF\_Responder/      SIF\_ObjectList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_Responders/SIF\_Responder/      SIF\_ObjectList/SIF\_Object/      SIF\_ExtendedQuerySupport | M |  | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
|  | SIF\_Responders/SIF\_Responder/      SIF\_ObjectList/SIF\_Object/      SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_Requesters | C | Encompasses all the requesters registered with this zone. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_Requesters/SIF\_Requester | MR |  |  |
| **@** | SourceId | M | The identifier of the SIF node that can request an object. This is the agent identifier that would appear in the SIF\_SourceId field of any SIF\_Header created by the agent. | [xs:token](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | SIF\_Requesters/SIF\_Requester/      SIF\_ObjectList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_Requesters/SIF\_Requester/      SIF\_ObjectList/SIF\_Object | MR |  |  |
| **@** | ObjectName | M | The name of the object being requested by this agent. | [ObjectNameType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#ObjectNameType) |
|  | SIF\_Requesters/SIF\_Requester/      SIF\_ObjectList/SIF\_Object/      SIF\_ExtendedQuerySupport | M |  | [xs:boolean](http://www.w3.org/TR/xmlschema-2/#boolean) |
|  | SIF\_Requesters/SIF\_Requester/      SIF\_ObjectList/SIF\_Object/      SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_SIFNodes/SIF\_SIFNode | MR |  |  |
| **@** | Type | M | The type of the node registered with the ZIS. Note that ZIS is forward-looking and not used currently; all information about this Zone/ZIS is contained outside SIF\_SIFNodes. | values:  Agent  ZIS |
|  | SIF\_SIFNodes/SIF\_SIFNode/      SIF\_Name | M | The descriptive name of the SIF node (i.e. Ramsey Food Services). | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString) |
|  | 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 image MIME type commonly supported by Web browsers (e.g. PNG, JPEG, GIF). Agents may optionally follow the more restrictive UUIDelines at [[FAVICON]](http://specification.sifinfo.org/Implementation/2.4/References.html#FAVICON). | [xs:anyURI](http://www.w3.org/TR/xmlschema-2/#anyURI) |
|  | SIF\_SIFNodes/SIF\_SIFNode/      SIF\_NodeVendor | O | The vendor of the SIF agent. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |
|  | 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 | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 32 | |
|  | 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. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |
|  | SIF\_SIFNodes/SIF\_SIFNode/      SIF\_Application/SIF\_Product | M | The name of the product that this agent supports. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 256 | |
|  | SIF\_SIFNodes/SIF\_SIFNode/      SIF\_Application/SIF\_Version | M | The version of the product. This field is informative only. | [xs:normalizedString](http://www.w3.org/TR/xmlschema-2/#normalizedString)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 32 | |
|  | 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](http://www.w3.org/TR/xmlschema-2/#token)   |  |  | | --- | --- | | [xs:maxLength](http://www.w3.org/TR/xmlschema-2/#rf-length) | 64 | |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Protocol) |
|  | SIF\_SIFNodes/SIF\_SIFNode/      SIF\_VersionList | M |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_EncryptionLevel) |
|  | 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](http://www.w3.org/TR/xmlschema-2/#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_SupportedProtocols/SIF\_Protocol | MR |  | [SIF\_Protocol](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Protocol) |
|  | SIF\_SupportedVersions | M | Enumerates the versions of the SIF Implementation Specification that this ZIS can use when communicating with the agent. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_SupportedVersions/SIF\_Version | MR | Lists a specific SIF Implementation Specification version. | [VersionType](http://specification.sifinfo.org/Implementation/2.4/CommonTypes.html#VersionType) |
|  | 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](http://www.w3.org/TR/xmlschema-2/#anyURI) |
|  | SIF\_Contexts | M |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_ServiceProviders | O |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ServiceProviders/SIF\_ServiceProvider | OR | 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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ServiceProviders/SIF\_ServiceProvider/      SIF\_ServiceList | M | The list of services provided by this node | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ServiceResponders/SIF\_ServiceResponder | OR |  |  |
| **@** | 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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ServiceResponders/SIF\_ServiceResponder/      SIF\_ServiceList | M | The list of services that will be responded to by this node. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#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](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_ServiceRequesters | OR | 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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ServiceRequesters/SIF\_ServiceRequester | OR |  |  |
| **@** | 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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ServiceRequesters/SIF\_ServiceRequester/      SIF\_ServiceList | M | The list of services that will be invoked by this node | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ServiceRequesters/SIF\_ServiceRequester/      SIF\_ServiceList/SIF\_Service/      SIF\_Contexts | O |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_ServiceSubscribers | O |  | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ServiceSubscribers/SIF\_ServiceSubscriber | OR |  |  |
| **@** | 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](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ServiceSubscribers/SIF\_ServiceSubscriber/      SIF\_ServiceList | M | The list of services that are subscribed to by this node. | [List](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ServiceSubscribers/SIF\_ServiceSubscriber/      SIF\_ServiceList/SIF\_Service | MR |  |  |
| **@** | ServiceName |  | The name of the SIF Zone Service as defined by a SIF Zone Service specification | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | 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](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#List) |
|  | SIF\_ServiceSubscribers/SIF\_ServiceSubscriber/      SIF\_ServiceList/SIF\_Service/      SIF\_Operations/SIF\_Operation | MR | A specific notification message that the agent is subscribed to | [xs:token](http://www.w3.org/TR/xmlschema-2/#token) |
|  | SIF\_ServiceSubscribers/SIF\_ServiceSubscriber/      SIF\_ServiceList/SIF\_Service/      SIF\_Contexts | O |  | [SIF\_Contexts](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html#SIF_Contexts) |
|  | SIF\_Metadata | O |  | [SIF\_Metadata](http://specification.sifinfo.org/Implementation/2.4/DataModelTaskForce.html#SIF_Metadata) |
|  | SIF\_ExtendedElements | O |  | [SIF\_ExtendedElements](http://specification.sifinfo.org/Implementation/2.4/DataModel.html#SIF_ExtendedElements) |

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\_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

# 7 Zone Services

## 7.1 Introduction

This section documents the set of SIF Zone Services which are defined using the service message types added to the SIF v2.4 infrastructure, and which leverage the advantages of the service paradigm (encapsulation of object data and behavior) in support of complex data exchange processes.

Zone Services is the third major conceptual component to be introduced into the SIF architecture. This introduction provides the necessary context for understanding them in relation to the Object Data Model and the SIF Infrastructure.

### 7.1.1 Important Terms

The following terms will be used throughout the rest of this section. Wherever possible, they reflect common industry usage and consensus.

#### 7.1.1.1 Service

A Service is a software application that responds to requests made of it by client applications. Any given application can be both a service used by multiple clients, and a client which itself uses other services.

Every service possesses a public interface defining exactly what operations may be initiated against the service. This interface specifies the operations the service supports, the data these operations accept and the results they return. Each service also has a private (unpublished) implementation that determines how the service will fulfill, or respond to, requests.

The fact that the service implementation is private (encapsulated) means that even if the details of that implementation radically change, the interface is unaffected and none of the clients of that service will be impacted. Ensuring that clients are independent of how a service is actually implemented is known as loose coupling, and a key enabler of the Zone Service architectural component.

#### 7.1.1.2 SIF Object Provider Service

A SIF Object Provider can be considered an “Object Service”. The Service interface in this case is composed completely of CRUD (Create, Read, Update, and Delete) operations for the object data it provides. Making a SIF Request is equivalent to invoking a “Read” operation, and the SIF Event equates to the “Create”, “Update” and “Delete” service operations.

Neither the structures of the SIF Object hierarchy nor the behavioral aspects of an educational process can be encapsulated by these Object Services.

#### 7.1.1.3 SIF Zone Service

The existing SIF infrastructure has been extended to support “non-CRUD” operations, which allow Zone Services to be constructed that encapsulate both the details of the object hierarchy and associated transactional behavior. Three SIF message types (ServiceInput, ServiceOutput and ServiceNotify) have been added to carry these non-CRUD operation requests, responses and event notifications respectively. As a result, any client of a Zone Service must support this extended SIF infrastructure.

All Zone Service interfaces will follow the “Document Literal” pattern. The payload of each of these message types will have its structure defined by an XML schema which can be validated using the corresponding schema definition supplied within this SIF specification.

Although Zone Services are “services” in the sense of encapsulating the details of the operations they perform for their clients, all communications with their clients MUST pass through the ZIS with the same communication mechanisms used by Object Clients and Object Providers. This provides the Zone Client to Zone Service link with the following “quality of service” (QoS) advantages:

* Messages are exchanged in a secure fashion (via message encryption and application authentication).
* A Zone Client or Service will never receive a message unless it has both been given permission to do so and the sender has been given the correct permission to provide it (typically via authorization from the Zone administrator).

This permission is granular, and a given client subscriber can be granted permission to receive none, some or all of the notification types provided by a Zone Service.

* Any legally posted message is guaranteed to be delivered, in order, once only, error free, or a notification of delivery failure will be given to the sender.
* A Zone Client need not know the location of its Zone Service. Any operation invocations for a named Zone Service will arrive there due to the content based routing capabilities provided by the ZIS.
* A Zone Client need not worry about allocating enough buffer space to receive an unexpectedly large response or notification from the Zone Service. The Zone Service infrastructure extensions support message packetization for all three service message types.
* A Zone Service need not know the location or the number of subscribers to its named notifications. When it issues a Notification, all legal subscribers are guaranteed to receive it, even if they are offline at the time of its publication, unless a total subscriber failure occurs. The Zone Service is not informed of these types of errors.

In brief, if the need for application interoperability is limited to sharing data, the solution should utilize existing client access to one or more Object Provider Services over the basic SIF infrastructure. However, if there is a need to support a more complex (possibly stateful) process choreography that requires multiple systems to coordinate their actions, one or more Zone Services should be employed.

### 7.1.2 New Message Types

Content-based routing for the SIF\_Request, SIF\_Event, SIF\_Response and SIF\_CancelRequests messages is based solely on the identified object type in the header of the message. Together those message types provide the client with the same CRUD interface (Create, Read, Update and Delete operations) to every type of object.

A given service type on the other hand may support a unique set of interface operations, which can affect multiple objects of varying types, in support of complex transactions or process choreographies. The existing SIF message types were incapable of being extended to support this additional functionality.

The three messages, which Zone Services use instead, are described below. All three have the name of the service and the operation being invoked as part of their message definition.

#### 7.1.2.1 Service Input

This message is issued by a Zone Service client to invoke a defined Zone Service operation. It is routed by the ZIS to the default provider for that service, and like the Request can be directed to a specific destination as specified in the SIF\_DestinationId in the ServiceInput header.

Service Inputs differ from Object Requests in the following ways:

* In addition to requesting object data, the ServiceInput message can also effect changes in one or more objects depending upon what elements are defined for the payload.
* Elements within the body of the ServiceInput can contain a detailed description of the changes (including state changes) that need to occur.
* The contents of the body of a Service Input message can range from no data at all to multiple elements, and are not restricted to elements from a single object.
* Service Input messages can be packetized if they are very large in size.

An example of a ServiceInput request sent by a client to the “Weather Service” requesting the forecast is shown in the [SIF\_ServiceInput](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ServiceInput) section.

#### 7.1.2.2 Service Output

This message is issued by a Zone Service in response to one of its operations being invoked by an arriving ServiceInput message, and is routed by the ZIS back to the issuing client. All internal implementation details of how the Zone Service processes the request to produce this response (potentially including sending multiple Change Events to a variety of Object Providers) are hidden from the client.

An example of the ServiceOutput response sent by the Weather Service in response to a ServiceInput request for the forecast is shown in the [SIF\_ServiceOutput](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ServiceOutput) section.

#### 7.1.2.3 Service Notify

This message is used to notify subscribers that a service related event has occurred, and can optionally provide a detailed description of exactly what that event was. Service Notifications differ from Object Events in the following ways:

* In addition to simple event notification, the Service Notification can also contain a detailed description of the changes (including state changes) that occurred across multiple objects.
* The contents of the body of a Service Notification message can range from no data at all to multiple elements, and are not restricted to elements from a single object.
* Service Notification messages can be packetized if they are very large (implying a subscriber to several notification types could be receiving interspersed packets from multiple Service Notification messages).
* Service Notifications may only be published by the Service for which they are defined.
* Multiple notification types may be defined for a given Service type, and all can be subscribed to separately.
* Service Notifications cannot be blocked by agents invoking SMB.

An example of a Service Notification of a Student Transfer issued by the “Food Service” Service is shown in the [SIF\_ServiceNotify](http://specification.sifinfo.org/Implementation/2.4/Infrastructure.html" \l "SIF_ServiceNotify) section.

### 7.1.3 SIF Design Requirements for Zone Service Technology

The following requirements apply to all Zone Services:

* All responses to invocations of Zone Service operations MUST be asynchronous.
* Zone Service operations MUST use a document-oriented messaging pattern and not RPC.
* Zone Services MAY expose any number of operations and Notification types.
* Zone Services MUST provision themselves in the Zone over the existing SIF Infrastructure.
* All Zone Service messages with large payloads exceeding the specified maximum buffer size MUST be packetized.

The following requirements apply to all Zone Service Clients:

* Clients MAY subscribe to individually named Notification types from the same Zone Service.
* Client access to Zone Service operations and Notifications MUST be governed by access control list (ACL) policies.

The following requirements, provided by the ZIS, apply to any zone in which Zone Services are deployed:

* Clients and Zone Service Providers MUST NOT communicate directly with each other.
* Zone Service Providers and their clients MUST be guaranteed the full quality of service (QOS) feature set provided by the SIF request/response protocol including:
  + Automatic message routing based on service name
  + Packet ordering
  + Guaranteed delivery or failure notification
  + Publish and subscribe message exchange pattern

### 7.1.4 Zone Service Design Best Practices

Best practices for designing a Zone Service include:

#### 7.1.4.1 Error Handling

* Faults SHOULD be defined for known exceptional cases.
* SIF\_Error MUST be used for error cases.
* Custom error codes MAY be defined for each service.

#### 7.1.4.2 Data Model Reuse

* The SIF Data Model SHOULD be used wherever possible.
* Service-specific data elements SHOULD NOT be used unless absolutely necessary.
* All non-custom objects exchanged during a Zone Service operation MUST also be available using Request/Response calls to an Object Provider.
* An optional object element MAY be mandatory in the service interface.
* A Service Notification SHOULD NOT report on changes to data within a single SIF Data Model object, as it would directly duplicate a normal SIF\_Event.
* Where applicable, service definitions can support the exchange of large quantities of data because all Zone Service messages support packetization.

#### 7.1.4.3 General Zone Service UUIDelines

* Any Zone Service interface whose operations update SIF Data Model object elements SHOULD be implemented by the Object Provider of those elements, to maintain the “single owner of record” for all SIF elements.
* Zone Service interfaces in the same “set” (ex: Student Record Exchange) SHOULD be implemented by a single application, to minimize deployment complexity.
* Zone Services SHOULD be stateless or use a state token where necessary.

### 7.1.5 Zone Service Usage Best Practices

Many interactions/transactions between participants in a business process are difficult to implement using a sequence of SIF\_Request/SIF\_Response and SIF\_Event messages. This can be true for a number of reasons.

* The data that needs to be exchanged consists of elements that span multiple object types and a complex choreography results between the Object Providers and the client.
* An application must be relied on to process data that it doesn’t own.

A special Object Event can be issued, but there is no standard way to detect if the application accepted the data since the subscriber to an Event does not respond to the publisher. So a complex exchange of Events with essentially the same elements duplicated in multiple “process objects” generally results.

* Multiple responses to a single request can occur.

While the second and subsequent responses can be issued as Events, there is no way to distinguish their differences since all events for a given object type are essentially requests to update the data in a specific object. A Zone Service supports multiple named Notifications which:

* + Are clearly distinguishable,
  + can be subscribed to separately,
  + and can contain whatever data is required (including state data).

Where one or more of the above situations is true, development of a Zone Service should be considered. Otherwise the primary design effort should be focused on defining additional data elements and/or SIF Data Model object types.