311-EMD-105

EOSDIS Maintenance and Development Project

Release 7.20
Spatial Subscription Server (SSS) Database Design and Schema Specifications for the EMD Project

Revision --

July 2007

Raytheon Company
Upper Marlboro, Maryland

Release 7.20 Spatial Subscription Server (SSS) Database Design and Schema Specifications for the EMD Project

Revision --

July 2007

Prepared Under Contract NAS5-03098
CDRL Item #023

RESPONSIBLE ENGINEER

Robert Hartranft 7/9/2007

Robert Hartranft Date
EOSDIS Maintenance and Development Project

SUBMITTED BY

Art Cohen 7/9/07

Art Cohen, Development Manager Date
EOSDIS Maintenance and Development Project

Raytheon Company
Upper Marlboro, Maryland

This page intentionally left blank.

Preface

This document is a formal contract deliverable. It requires Government review and approval within 45 business days. Changes to this document will be made by document change notice (DCN) or by complete revision.

Any questions should be addressed to:

Data Management Office
The EMD Project Office
Raytheon Company
1616 McCormick Drive
Upper Marlboro, Maryland 20774-5301

Revision History

Document Number	Status/Issue	Publication Date	CCR Number
311-EMD-105	Original	July 2007	07-0337

This document describes the data design and database specification for the Subscription Server subsystem. It is one of eleven documents comprising the detailed database design specifications for each of the EMD subsystems.

The subsystem database design specifications for the as delivered system include:

311-EMD-100	Release 7.20 INGEST (INS) Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-101	Release 7.20 Science Data Server Database Design and Schema Specifications for the EMD Project
311-EMD-102	Release 7.20 Storage Management and Data Distribution Subsystems Database Design and Schema Specifications for the EMD Project
311-EMD-103	Release 7.20 Systems Management Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-104	Release 7.20 Order Manager Database Design and Schema Specifications for the EMD Project
311-EMD-105	Release 7.20 Spatial Subscription Server (SSS) Database Design and Schema Specifications for the EMD Project
311-EMD-106	Release 7.20 Data Pool Database Design and Schema Specifications for the EMD Project

Entity Relationship Diagrams (ERDs) presented in this document have been exported directly from tools and some cases contain too much detail to be easily readable within hard copy page constraints. The reader is encouraged to view these drawings on-line using the Portable Document Format (PDF) electronic copy available via the ECS Data Handling System (EDHS) on the world-wide web at http://edhs1.gsfc.nasa.gov.

Abstract

This document outlines Release 7.20 “as-built” database design and database schema of the Spatial Subscription Server database including the physical layout of the database and initial installation parameters.

Keywords: data**,** database, design, configuration, database installation, scripts, security, data model, data dictionary, replication, performance tuning, SQL server, database security, replication, database scripts

This page intentionally left blank.

Contents

Preface

Abstract

Contents

1. Introduction

2. Related Documents

3. Data Design

4. Performance and Tuning Factors

5. Database Security

6. Scripts

List of Figures

List of Tables

Appendix A. Entity Relationship Diagram

Abbreviations and Acronyms

This page intentionally left blank.

1. Introduction

1.1 Purpose and Scope

The purpose of Spatial Subscription Server (SSS) Database Design and Database Schema Specification document is to describe the database design and schema specifications implemented to support the data requirements of SSS Release 7.20 CSCI.

1.2 Document Organization

Section 1 provides information regarding the identification, purpose, scope and audience of this document.

Section 2 provides a listing of the related documents, which were used as a source of information for this document.

Section 3 contains the database overview for the SSS physical data model which includes a description of the database tables, triggers, stored procedures, flat files, and attributes.

Section 4 provides a description of database performance and tuning factors such as indexes, caches, and segments.

Section 5 provides a description of the database security infrastructure used and list of the users, groups, roles, and permissions available upon initial installation.

Section 6 provides a description of scripts used for installation, de-installation, backup/recovery, and other miscellaneous functions.

This page intentionally left blank.

2. Related Documents

2.1 Applicable Documents

The following documents, including Internet links, are referenced in this document, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume.

The following documents, including Internet links, are referenced in this document, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume.

305-EMD-100	Release 7.20 Segment Design Specification for the EMD Project
920-TDN-009	DAAC Hardware Database Mapping/NSIDC
920-TDE-009	DAAC Hardware Database Mapping/EDC
920-TDL-009	DAAC Hardware Database Mapping/LARC
920-TDS-009	DAAC Hardware Database Mapping/SMC
920-TDN-010	DAAC Database Configuration/NSIDC
920-TDE-010	DAAC Database Configuration/EDC
920-TDL-010	DAAC Database Configuration/LARC
920-TDS-010	DAAC Database Configuration/SMC
920-TDN-011	DAAC Sybase Log Mapping/NSIDC
920-TDE-011	DAAC Sybase Log Mapping/EDC
920-TDL-011	DAAC Sybase Log Mapping/LARC
920-TDS-011	DAAC Sybase Log Mapping/SMC
922-TDN-013	Disk Partitions/NSIDC
922-TDE-013	Disk Partitions/EDC
922-TDL-013	Disk Partitions/LARC
922-TDS-013	Disk Partitions/SMC

These documents are maintained as part of the ECS baseline and available on the world wide web at the URL: http://cmdm.east.hitc.com/baseline. Please note that this is a partial mirror site in that some items are not available (they are identified) since this is OPEN to all. This site may also be reached through the EDHS homepage. Scroll page to the connections line and click on the EMD Baseline Information System link.

2.2 Information Documents

The following documents, although not directly applicable, amplify or clarify the information presented in this document. These documents are not binding on this document.

609-EMD-100	Release 7.20 Operations Tools Manual for the EMD Project
611-EMD-100	Release 7.20 Mission Operation Procedures for the EMD Project

3. Data Design

3.1 Database Overview

The SSS database implements the large majority of the persistent data requirements for the SSS subsystem. The database is designed in such a manner as to satisfy business policy while maintaining data integrity and consistency. Database tables are implemented using the Sybase Relational Database Management System (RDBMS). All components of the SSS database are described in the section which follow, in sufficient detail to support maintenance needs.

3.1.1 Physical Data Model Entity Relationship Diagram

The Entity Relationship Diagram (ERD) presents a schematic depiction of the SSS physical data model. The ERDs presented here for the SSS database were produced using the Power Designor Data Architect Computer Aided Software Engineering (CASE) tool. ERDs represent the relationship between entities or database tables. On ERDs, tables are represented by rectangles and relationships are represented as arrow (see Figure 3-1).

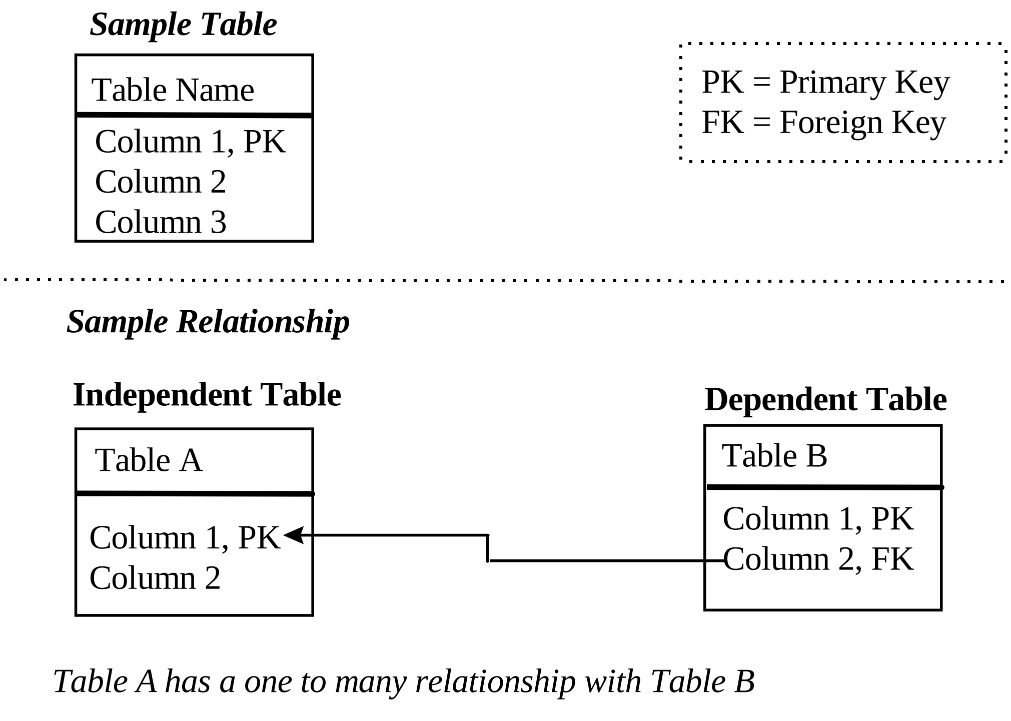

Figure 3-1. ERD Key

3.1.2 Tables

A listing the tables in the SSS database is given here. A brief definition of each of these tables follows including a listing of the columns comprising the table in Table 3-1. The column list indicates if the column is part of the primary key for the table, that is, if the columns can be used alone or in combination with other primary key columns to uniquely identify a single row in the table. The column list also indicates whether the column is a mandatory attribute that must be included in every row.

The set of tables can be partitioned into four logical groupings:

Subscriptions – information about subscriptions, their qualifiers, or their associated actions

Events – information about granule events to be matched with subscriptions

Actions – information about matched subscriptions whose actions are being executed

Miscellaneous – other tables, pertaining to configuration parameters or database cleanup

Table 3-1. Data Tables Listing (1 of 2)

Table Name	Logical Grouping
EcDbDatabaseVersions	Database Versioning Information
EcNbActionDefinition	Subscriptions
EcNbActionQueue	Actions
EcNbActionQueueFront	Actions
EcNbActionQueueLog	Actions
EcNbActionQueueRear	Actions
EcNbDistribution	Actions
EcNbDpEventDetails	Events
EcNbDuplicateDplInserts	Events
EcNbEventAttrXref	Events
EcNbEventDefinition	Events
EcNbEventDefinitionObsolete	Events
EcNbEventMetadataAttrDef	Events
EcNbEventMetadataDate	Events
EcNbEventMetadataFloat	Events
EcNbEventMetadataString	Events
EcNbEventMetadataInteger	Events
EcNbEventMetadataNose	Events
EcNbEventTruth	Events
EcNbMatchingExpNextId	Subscriptions
EcNbMatchingExpression	Subscriptions
EcNbNoseMatchingExpression	Subscriptions
EcNbNotificationAction	Subscriptions

Table 3-1. Data Tables Listing (2 of 2)

Table Name	Logical Grouping
EcNbDpAction	Subscriptions
EcNbOrderAction	Subscriptions
EcNbPotentialMatches	Miscellaneous
EcNbQueueInfo	Miscellaneous
EcNbSbConfiguration	Miscellaneous
EcNbSpatialMatchingExpression	Subscriptions
EcNbSubEventQueueFront	Events
EcNbSubEventQueueLog	Events
EcNbSubEventQueueRear	Events
EcNbSubMatchExp_XREF	Subscriptions
EcNbSubMatchingExpDate	Subscriptions
EcNbSubMatchingExpFloat	Subscriptions
EcNbSubMatchingExpInteger	Subscriptions
EcNbSubMatchingExpString	Subscriptions
EcNbSubscribedEventQueue	Events
EcNbSubscription	Subscriptions
EcNbSubscriptionNextId	Subscriptions
EcNbActionNextId	Subscriptions
EcNbDeleteRequestQueue	Miscellaneous
EcNbDeleteRequestQueueLock	Miscellaneous

Table 3-2 identifies the current version level of the SSS database.

Table 3-2. EcDbDatabaseVersions

Column Name	Data Type	PK Column	Mandatory Column
EcDbSchemaVersionId	smallint	Yes	Yes
EcDbDropVersion	char(64)	Yes	Yes
EcDbDropDescription	varchar(255)	No	No
EcDbCurrentVersionFlag	char(1)	No	No
EcDbDatabaseName	varchar(255)	No	No
EcDbDropInstallDate	datetime	No	No
EcDbSybaseVersion	varchar(255)	No	No
EcDbSybaseServer	varchar(255)	No	No
EcDbComments	varchar(255)	No	No
EcDbUpdateProcess	varchar(255)	No	No

Table 3-3 identifies the actions that are associated with a subscription. A non-bundled subscription may have up to three actions (acquire, email, data pool), and each action would be represented by a row in this table.

Table 3-3. EcNbActionDefinition

Column Name	Data Type	PK Column	Mandatory Column
actionId	int	Yes	Yes
subscriptionId	int	No	Yes
actionType	char(10)	No	No
Act_qPriority	int	No	No

Table 3-4 contains the next available identifier for an action.

Table 3-4. EcNbActionNextId

Column Name	Data Type	PK Column	Mandatory Column
nextId	int	No	Yes

Table 3-5 represents the queue of matched subscriptions to be processed by the action driver.

Table 3-5. EcNbActionQueue

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
ActionId	int	Yes	Yes
actionQueueId	int	No	Yes
Act_qPriority	int	No	No
eventId	int	No	No
subscriptionId	int	No	Yes
granUR	varchar(255)	No	No
failedActionTaken	char(1)	No	No
updateTime	datetime	No	No
retry	int	No	Yes
deleteRequest	varchar(1)	No	Yes

Table 3-6 keeps track of the next subscription action to be dequeued from the EcNbActionQueue.

Table 3-6. EcNbActionQueueFront

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
qFrontId	int	No	No
qLogicalLock	int	No	No

Table 3-7 contains audit trail data for actions that have been processed.

Table 3-7. EcNbActionQueueLog

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
ActionId	int	Yes	Yes
actionId	int	No	Yes
actionStatus	varchar(30)	Yes	Yes
actionDateTime	datetime	No	No
pid	int	No	No

Table 3-8 keeps track of the next available position in the EcNbActionQueue.

Table 3-8. EcNbActionQueueRear

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
qRearId	int	No	No

Table 3-9 contains references to event and action data that are ready for cleanup by the deletion driver.

Table 3-9. EcNbDeleteRequestQueue

Column Name	Data Type	PK Column	Mandatory Column
deleteRequestId	numeric(22)	Yes	Yes (identity)
qPriority	int	No	Yes
objectId	int	No	Yes
objectType	char(1)	No	Yes
objectCompleteTime	datetime	No	Yes
deleteRequestStatus	tinyint	No	Yes

Table 3-10 provides data to support logical locking while dequeuing EcNbDeleteRequestQueue.

Table 3-10. EcNbDeleteRequestQueueLock

Column Name	Data Type	PK Column	Mandatory Column
lLock	int	No	Yes

Table 3-11 keeps track of distribution requests for a particular granule and user. Its purpose is to ensure that the same granule is not distributed more than once to a particular user. It is written to by the action driver. A duplicate entry will result in a duplicate key error at runtime, which is handled by the action driver. The deletion driver performs cleanup.

Table 3-11. EcNbDistribution

Column Name	Data Type	PK Column	Mandatory Column
eventId	int	Yes	Yes
distrString	varchar(255)	Yes	Yes

Table 3-12 contains the data for each data pool action.

Table 3-12. EcNbDpAction

Column Name	Data Type	PK Column	Mandatory Column
actionId	int	Yes	Yes
retentionPeriod	int	No	Yes
retentionPriority	int	No	Yes
insertMetadataOnly	char(1)	No	Yes

Table 3-13 is a staging area for DataPool actions waiting to be inserted into the Data Pool action queue in the Data Pool database. An insert trigger on this table performs the actual insert into the Data Pool.

Table 3-13. EcNbDpEventDetails

Column Name	Data Type	PK Column	Mandatory Column
actionId	int	Yes	Yes
subscriptionId	int	Yes	Yes
dbId	Numeric(16)	Yes	Yes
eventDPactions	int	No	No
ShortName	varchar(8)	No	No
esdtVersion	int	No	No
retentionPriority	int	No	No
retentionPeriod	int	No	No
insertMetadataOnly	char(1)	No	No
themeID	int	No	No
dispatchPriority	int	No	No

Table 3-14 contains reports on any attempts to insert the same (ecsId, subId) pair more than once into the DlInsertActionQueue table in the DataPool database.

Table 3-14. EcNbDuplicateDplInserts

Column Name	Data Type	PK Column	Mandatory Column
ecsId	numeric(16)	Yes	Yes
subId	int	Yes	Yes
timestamp	datetime	No	No
counter	int	No	No

Table 3-15 cross-references valid ESDTs with valid metadata attributes

Table 3-15. EcNbEventAttrXref

Column Name	Data Type	PK Column	Mandatory Column
EventType	varchar(80)	Yes	Yes
ESDT_Id	varchar(8)	Yes	Yes
VersionID	int	Yes	Yes
attributeName	varchar(255)	Yes	Yes
attributeType	varchar(50)	Yes	Yes

Table 3-16 holds information about the different types of media on which data will come in.

Table 3-16. EcNbEventDefinition

Column Name	Data Type	PK Column	Mandatory Column
EventType	varchar(80)	Yes	Yes
ESDT_Id	varchar(8)	Yes	Yes
VersionID	int	Yes	Yes
eventID	int	No	No

Table 3-17 ESDTs that have been withdrawn from use by subscribers

Table 3-17. EcNbEventDefinitionObsolete

Column Name	Data Type	PK Column	Mandatory Column
EventType	varchar(80)	Yes	Yes
ESDT_Id	varchar(8)	Yes	Yes
VersionID	int	Yes	Yes
timeWithdrawn	datetime	Yes	Yes

Table 3-18 defines the event’s metadata attributes.

Table 3-18. EcNbEventMetadataAttrDef

Column Name	Data Type	PK Column	Mandatory Column
attributeName	varchar(255)	Yes	Yes
attributeType	varchar(50)	Yes	Yes
sdsrvTableName	varchar(50)	No	Yes
sdsrvColumnName	varchar(50)	No	Yes
sdsrvParameterName	varchar(40)	No	No
parameterColumnName	varchar(50)	No	No
attributeFlag	char(1)	No	No

Table 3-19 stores the values of all date attributes of the granule associated with the event.

Table 3-19. EcNbEventMetadataDate

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
eventId	int	Yes	Yes
attributeName	varchar(255)	Yes	Yes
attributeType	varchar(50)	Yes	Yes
EventMetadataValueDate	datetime	No	No

Table 3-20 stores the values of all floating point attributes of the granule associated with the event.

Table 3-20. EcNbEventMetadataFloat

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	No	Yes
eventId	int	No	Yes
attributeName	varchar(255)	No	Yes
attributeType	varchar(50)	No	Yes
eventValueFloat	float	No	No

Table 3-21 stores the values of all integer attributes of the granule associated with the event.

Table 3-21. EcNbEventMetadataInteger

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
eventId	int	Yes	Yes
attributeName	varchar(255)	Yes	Yes
attributeType	varchar(50)	Yes	Yes
eventValueInteger	int	Yes	Yes

Table 3-22 contains the metadata for an event whose associated spatial type is orbit polygon. (NOSE = Nominal Orbit Spatial Extent)

Table 3-22. EcNbEventMetadataNose

Column Name	Data Type	PK Column	Mandatory Column
attributeName	varchar(255)	Yes	Yes
attributeType	varchar(50)	Yes	Yes
qPriority	int	Yes	Yes
eventId	int	Yes	Yes
PathNo	smallint	Yes	Yes
StartBlock	smallint	Yes	Yes
EndBlock	smallint	No	Yes
platInstCode	tinyint	yes	Yes

Table 3-23 stores the values of all string attributes of the granule associated with the event.

Table 3-23. EcNbEventMetadataString

Column Name	Data Type	PK Column	Mandatory Column
sequenceNumber	numeric(9)	Yes	Yes
qPriority	int	No	Yes
eventId	int	No	Yes
attributeName	varchar(255)	No	Yes
attributeType	varchar(50)	No	Yes
EventValue	varchar(255)	No	No

Table 3-24 contains the list of matching expressions that are true for an event. Used in subscription resolution.

Table 3-24. EcNbEventTruth

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
eventId	int	Yes	Yes
matchingExpressionId	int	Yes	Yes

Table 3-25 contains the next matching ExpressionId to be created.

Table 3-25. EcNbMatchingExpNextId

Column Name	Data Type	PK Column	Mandatory Column
nextId	int	No	Yes

Table 3-26 contains a unique list of subscription qualifiers base class information. It is cross-referenced with the EcNbSubscription table through the EcNbSubMatchExp_XREF table.

Table 3-26. EcNbMatchingExpression

Column Name	Data Type	PK Column	Mandatory Column
matchingExpressionId	int	Yes	Yes
attributeName	varchar(255)	No	Yes
attributeType	varchar(50)	No	Yes
negationFlag	char(1)	No	No

Table 3-27 contains a list of qualifiers for subscriptions spatially qualified by orbit polygon data.

Table 3-27. EcNbNoseMatchingExpression

Column Name	Data Type	PK Column	Mandatory Column
matchingExpressionId	int	Yes	Yes
PathNo	smallint	Yes	Yes
StartBlock	smallint	Yes	Yes
EndBlock	smallint	No	Yes
platInstCode	tinyint	Yes	Yes

Table 3-28 contains the email address for recipients of subscription notifications. It also contains a flag to indicate what type of metadata should be included in the notification.

Table 3-28. EcNbNotificationAction

Column Name	Data Type	PK Column	Mandatory Column
actionId	int	Yes	Yes
notficationEmailAddr	varchar(255)	No	No
metadataFlag	char(1)	No	No
emailUserString	varchar(255)	No	No

Table 3-29 contains acquire data associated with a subscription.

Table 3-29. EcNbOrderAction

Column Name	Data Type	PK Column	Mandatory Column
actionId	int	Yes	Yes
actionPriority	varchar(10)	No	Yes
userProfile	varchar(30)	No	Yes
emailAddress	varchar(255)	No	Yes
mediaFormat	varchar(20)	No	Yes
mediaType	varchar(20)	No	Yes
notifyType	varchar(20)	No	No
ftpUser	varchar(30)	No	No
ftpPassword	varchar(255)	No	No
destinationNode	varchar(255)	No	No
destinationDirectory	varchar(255)	No	No
userstring	varchar(255)	No	No

Table 3-30 is a work table used by the event driver to track subscriptions that are potential matches to the current event.

Table 3-30. EcNbPotentialMatches

Column Name	Data Type	PK Column	Mandatory Column
pid	int	No	No
subscriptionId	int	No	No

Table 3-31 is a list of valid event queue and action queue priorities. For the initial release, it will consist of a single row with value 1.

Table 3-31. EcNbQueueInfo

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes

Table 3-32 holds time lapse configuration parameters for use by the deletion driver and the recovery driver.

Table 3-32. EcNbSbConfiguration

Column Name	Data Type	PK Column	Mandatory Column
configurationParm	varchar(25)	No	Yes
configurationValueInt	int	No	Yes

Table 3-33 holds subscription qualification information for spatial attributes.

Table 3-33. EcNbSpatialMatchingExpression

Column Name	Data Type	PK Column	Mandatory Column
matchingExpressionId	int	Yes	Yes
llat	float	No	Yes
llong	float	No	Yes
ulat	float	No	Yes
ulong	float	No	Yes
spatialConstraint	llbox	No	Yes

Table 3-34 keeps track of the next subscribed event to be dequeued from the EcNbSubscribedEventQueue.

Table 3-34. EcNbSubEventQueueFront

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
qFrontId	int	No	No
qLogicalLock	int	No	No

Table 3-35 contains audit trail data for events being processed.

Table 3-35. EcNbSubEventQueueLog

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
eventId	int	Yes	Yes
eventStatus	varchar(30)	Yes	Yes
subscriptionId	int	Yes	Yes
eventDateTime	datetime	No	No
pid	int	No	No

Table 3-36 keeps track of the next available position in EcNbSubscribedEventQueue.

Table 3-36. EcNbSubEventQueueRear

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
qRearId	int	No	No

Table 3-37 cross-references the subscription and matching expression tables.

Table 3-37. EcNbSubMatchExp_XREF

Column Name	Data Type	PK Column	Mandatory Column
subscriptionId	int	Yes	Yes
matchingExpressionId	int	Yes	Yes

Table 3-38 holds subscription qualifications for date attributes.

Table 3-38. EcNbSubMatchingExpDate

Column Name	Data Type	PK Column	Mandatory Column
matchingExpressionId	int	Yes	Yes
LowerBoundD	datetime	No	No
UpperBoundD	datetime	No	No

Table 3-39 holds subscription qualifications for floating point attributes.

Table 3-39. EcNbSubMatchingExpFloat

Column Name	Data Type	PK Column	Mandatory Column
matchingExpressionId	int	Yes	Yes
LowerBoundF	float	No	No
UpperBoundF	float	No	No

Table 3-40 holds subscription qualifications for integer attributes.

Table 3-40. EcNbSubMatchingExpInteger

Column Name	Data Type	PK Column	Mandatory Column
matchingExpressionId	int	Yes	Yes
LowerBoundI	int	No	No
UpperBoundI	int	No	No

Table 3-41 holds subscription qualifications for string attributes.

Table 3-41. EcNbSubMatchingExpString

Column Name	Data Type	PK Column	Mandatory Column
matchingExpressionId	int	Yes	Yes
LowerBoundS	varchar(255)	No	No
UpperBoundS	varchar(255)	No	No

Table 3-42 represents the queue of incoming events that are possible matches to subscriptions

Table 3-42. EcNbSubscribedEventQueue

Column Name	Data Type	PK Column	Mandatory Column
qPriority	int	Yes	Yes
eventId	int	Yes	Yes
eventQueueId	int	No	Yes
EventType	varchar(80)	No	Yes
ESDT_Id	varchar(8)	No	Yes
VersionID	int	No	Yes
dbID	numeric(16)	No	No
granUR	varchar(255)	No	No
enqueueDateTime	datetime	No	No

Table 3-43 contains basic data about subscriptions that have been entered.

Table 3-43. EcNbSubscription

Column Name	Data Type	PK Column	Mandatory Column
subscriptionId	int	Yes	Yes
EventType	varchar(80)	No	Yes
ESDT_Id	varchar(8)	No	Yes
VersionID	int	No	Yes
userId	varchar(14)	No	Yes
NumMatchExps	int	No	Yes
status	varchar(10)	No	Yes
startDate	datetime	No	Yes
expirationDate	datetime	No	Yes
logicalOp	char(3)	No	Yes
bundlingOrderId	varchar(10)	No	No
oldExpirationDate	datetime	No	No
themeName	varchar(40)	No	No
updateTime	datetime	No	No
NumNonSpatial	int	No	No

Table 3-44 contains the next subscriptionId to be created.

Table 3-44. EcNbSubscriptionNextId

Column Name	Data Type	PK Column	Mandatory Column
nextId	int	No	Yes

3.1.3 Columns

Brief definitions of each of the columns present in the database tables defined above are contained in Table 3-45.

Table 3-45. Column Descriptions (1 of 6)

Column Name	**Column Description**	**Valid Values**
Act_qPriority	The priority associated with the action at subscription entry time	For the initial release this value will always be 1
actionDateTime	A timestamp associated with the log entry	
actionId	A numerical identifier for a particular Priority associated with the action action	
ActionId	Original action queue identifier for actions that have been requeued; same as the current ID if action has not been requeued	
actionPriority	Priority associated with the action	XPRESS, VHigh, VHIGH, HIGH, NORMAL, LOW
actionQueueId	Current identifier for the queue	
actionStatus	The current state of the action	Action Enqueue Action Dequeue Acquire Action Notification Finished Action Processing
actionType	The type of action: Notify, Acquire, DataPool	
attributeFlag	An indicator of the attribute type	P – PSA M – measured S – spatial
attributeName	The name of the attribute	
attributeType	The type of attribute (column in the following tables: EcNbEventMetadataAttrDef EcNbEventMetadataDate EcNbEventMetadataFloat EcNbEventMetadataString EcNbEventMetadataNose EcNbEventMetadataInteger) Attribute type for the matching expression (attributeType column in EcNbMatchingExpression table)	int, float, datetime, varchar, llbox, gpolygon, PathBlock
bundlingOrderId	If the subscription is bundled, the MSS order ID of the bundling order	
configurationParm	Name of the configuration parameter	
configurationValueInt	Value of the configuration parameter expressed as a number of seconds	
counter	The number of times an insert was attempted for this row	
dbId	Granule identifier in the SDSRV database	
dbID	Granule identifier in the SDSRV database	
deleteRequest	Flag indicating whether operator wishes to delete this action from the action queue	“Y”, “N” (default)

Table 3-45. Column Descriptions (2 of 6)

Column Name	Column Description	Valid Values
deleteRequestId	Identifier for an entry in this queue	
deleteRequestStatus	Current state of deletion processing for the object	
destinationDirectory	Directory where data is to be pushed	
destinationNode	Machine that is the target for an FtpPush distribution	
distrString	For a bundled subscription, this is the bundling order ID string. For a non-bundled subscription, this is the userId associated with the acquire action	
dispatchPriority	Part of a Data Pool action definition for a subscription that is inserted into the Data Pool for use by that subsystem	1-255. 200 is the default.
EcDbComments	Notes or comments on the database version level	
EcDbCurrentVersionFlag	Flag indicating if this row represents the current database version entry	
EcDbDatabaseName	The name of the database for which this database versions level is applied	
EcDbDropDescription	The official name of the ECS software drops for this database version level	
EcDbDropInstallDate	The date and time that the database versions level was installed	
EcDbDropVersion	The official description of the ECS software drops for this database version level	
EcDbSchemaVersionId	The subsystem-specific identifier for this database schema version	
EcDbSybaseServer	The name of the baseline Sybase SQL server controlling this database	
EcDbSybaseVersion	The software release version of the Sybase SQL server in place when this database version level was initially installed	
EcDbUpdateProcess	The installation method by which this database version level was installed	
ecsId	The identifier used for the granule by the Science Data Server	
emailAddress	Email address of the user for distribution notification	
emailUserString	An optional string supplied by the user to be incorporated in email	
EndBlock	Ending block number for the path	
enqueueDateTime	The date and time an event was inserted into the subscribed event queue	
ESDT_Id	The short name of the ESDT associated with the event	

Table 3-45. Column Descriptions (3 of 6)

Column Name	Column Description	Valid Values
esdtVersion	Version ID of the associated ESDT	
eventDateTime	Timestamp for the log entry	
eventDPactions	Total number of Data Pool actions associated with the event	
eventId	The identifier for the event associated with the subscription	
eventID	The identifier associated with the event in the SDSRV database. A value of zero denotes a withdrawn event marked for deletion	
EventMetadataValueDate	The datetime value associated with the granule	
eventQueueId	Event queue position	
eventStatus	Current state of processing for the event	
EventType	The type of even	INSERT DELETE UPDATEMETADATA
EventValue	String value for the metadata	
eventValueFloat	Float value of the attribute	
eventValueInteger	Integer value of the attribute	
expirationDate	Date after which the subscription expires	
failedActionTaken	Indicates that operator has approved the removal of the failed action from the system	
ftpPassword	User's password for an FtpPush distribution	
ftpUser	Name of user for an FtpPush distribution	
granUR	The UR for the granule associated with the event	
insertMetadataOnly	A ▯ oolean indicating whether the Data Pool action is for inserting both the science and metadata files or only the metadata file for that granule in the Data Pool directories	'Y' – means metadata only 'N' – means both science and metadata
llat	south latitude for llbox	
llong	west longitude for llbox	
logicalOp	Indicates whether multiple qualifiers for a subscription are to be logically combined using AND or OR	AND (default), OR
lLock	Used for queue locking	
LowerBoundD	Earliest date in the date range	
LowerBoundF	Minimum float value for the expression	
LowerBoundI	Minimum integer value for the expression	
LowerBoundS	Minimum string value for the expression	

Table 3-45. Column Descriptions (4 of 6)

Column Name	Column Description	Valid Values
matchingExpressionId	Identifier of a subscription qualifier satisfied by the event (column in EcNbEventTruth table) Identifier for the matching expression for the column matchingExpressionId in other tables	
mediaFormat	Format for the data to be acquired	FILEFORMAT
mediaType	Type of distribution to be performed	FtpPush FtpPull scp
metadataFlag	A oolean indicating whether all metadata should be included in email or only include names and values for metadata attributes associated with the subscription qualifiers	Y – means all metadata N – means qualifying metadata only
negationFlag	Indicates whether match is by equality	(blank character) – equality (non-blank character) – inequality
nextId	Next available identifier for a matching expression	
notficationEmailAddr	Email address of recipient	
notifyType	Type of distribution notification	MAIL
NumMatchExps	Number of subscription qualifiers	
NumNonSpatial	Number of non-spatial subscription qualifiers	
objectId	The event id or action to be deleted	
objectType	Type of object slated for deletion	'E' (event); 'S' (subscription action)
objectCompleteTime	Time the object was queued for deletion	
oldExpirationDate	Expiration date originally entered for a subscription that has been superseded by the expiration date of a bundling order	
parameterColumnName	The column name associated with the attribute definition in SDSRV	
PathNo	Path number associated with the orbit	
pid	The UNIX process ID of the action driver that dequeued this action	
platInstCode	Platform instrument code	
qFrontId	A pointer to the front of the queue	
qLogicalLock	Not used, except to hold a lock on the table during a transaction	
qPriority	The priority associated with the action	
qRearId	A pointer to the rear of the queue	
retentionPeriod	Retention period in the Data Pool database (see documentation for the table DlInsertActionQueue in the Data Pool database)	

Table 3-45. Column Descriptions (5 of 6)

Column Name	Column Description	Valid Values
retentionPriority	Retention priority in the Data Pool database (see documentation for the table DlInsertActionQueue in the Data Pool database)	
retry	Number of times an action has been retried after failing	Default Value is 0
sdsrvColumnName	The table in the SDSRV database where the attribute value appears	
sdsrvParameterName	A numeric string associated with the attribute definition in SDSRV	
sdsrvTableName	Name of the table in the Science Data Server database where this metadata is located	
sequenceNumber	An identity that serves as the primary key of the table EcNbEventMetadataString	
ShortName	The short name of an ESDT	
spatialConstraint	The LLBOX used to spatially qualify a subscription	
StartBlock	The starting block number of orbit data associated with a granule	
startDate	The date when a subscription takes effect	
status	Subscription status	Active, Inactive, Canceled
subscriptionId	The identifier of the subscription associated with this action	
subId	The subscription identifier inserted into the DataPool insert action queue	
themeID	This is the themeId from the Data Pool database in the case where the subscription's Data Pool action has an associated theme enabled for insert. This value can be NULL	
themeName	The name of a Data Pool theme, if the subscription has a data pool action associated with a theme	
timeWithdrawn	The date and time the ESDT was withdrawn from use by the subscribers.	
timestamp	The time when the row was inserted into the table.	
ulat	North latitude for llbox	
ulong	East longitude for llbox	
updateTime	Time of last update	
UpperBoundD	Latest date in the date range	
UpperBoundF	Minimum float value for the expression	
UpperBoundI	Maximum integer value for the expression	
UpperBoundS	Maximum string value for the expression	
userId	Name of the owner of the subscription	
userProfile	User profile name (as it appears in the MSS accounting database)	

Table 3-45. Column Descriptions (6 of 6)

Column Name	Column Description	Valid Values
userstring	An optional string defined by the user for secondary identification	
VersionID	The version number of the ESDT associated with the event	

3.1.4 Domains

Sybase supports the definition of specific data types, domains, to further limit the format of data for given column. User-defined data types are no longer used in the SSS database.

3.1.5 Rules

Sybase supports the definitions of rules. Rules provide a means for enforcing domain constraints on a given column. Multiple rules may be defined for a given column. Multiple rules are not always uniquely named. The following rules are defined:

Rules	Description
RuleMediaType	(‘FtpPush’, ‘FtpPull’, ‘scp’) bound to EcNbOrderAction.mediaType.
RuleYesNo	(‘Y’,’N’) bound to EcNbDpAction.insertMetadataOnly and EcNbNotificationAction.metadataFlag.
RuleDistPriority	(‘VHigh’, ‘VHIGH’, ‘HIGH’, ‘NORMAL’, ‘LOW’, ‘XPRESS’) bound to EcNbOrderAction.actionPriority.
RuleMediaFormat	(‘FILEFORMAT’) bound to EcNbOrderAction.mediaFormat.
RuleNotifyType	(‘MAIL’) bound to EcNbOrderAction.notifyType.
RuleLogicalOp	(‘AND’, ‘OR’) bound to EcNbSubscription.logicalOp.

3.1.6 Defaults

Defaults are used to supply a value for a column when one is not defined at insert time. The defaults defined in Sybase for the SSS database are described herein.

Column Default	Default Value
EcNbSubscription.NumMatchExps	0
EcNbSubscription.NumMatchExps	0
EcNbSubscription.status	Active
EcNbSubscription.startDate	01/01/2000
EcNbActionDefinition.Act_qPriority	1
EcNbActionQueue.retry	0
EcNbActionQueue.deleteRequest	N

3.1.7 Views

Sybase allows the definition of views as a means of limiting an application or users access to data in a table or tables. Views create a logical table from columns found in one or more tables.

View	Description
EcNbSubscriptionDataPool	It is used by the SSS GUI to list subscriptions which have associated Data Pool actions

3.1.8 Integrity Constraints

Sybase allows the enforcement of referential integrity via the use of declarative integrity constraints. Integrity constraints allow the SQL server to enforce primary and foreign key integrity checks automatically without requiring programming. Sybase constraints support "restrict-only" operations. This means that a row cannot be deleted or updated if there are rows in other tables having a foreign key dependency on that row. Cascade delete and update operations can not be performed if a declarative constraint has been used. All declarative integrity constraints defined in the SSS database are described in Tables 3-46 through 3-53.

Table 3-46. Dependencies on Table: EcNbMatchingExpression

Referenced by	Primary Key	Referenced by
EcNbSubMatchingExpFloat	matchingExpressionId	matchingExpressionId
EcNbSubMatchingExpInteger	matchingExpressionId	matchingExpressionId
EcNbSubMatchingExpString	matchingExpressionId	matchingExpressionId
EcNbSubMatchingExpDate	matchingExpressionId	matchingExpressionId
EcNbSpatialMatchingExpression	matchingExpressionId	matchingExpressionId
EcNbNoseMatchingExpression	matchingExpressionId	matchingExpressionId
EcNbSubMatchExp_XREF	matchingExpressionId	matchingExpressionId

Table 3-47. Dependencies on Table: EcNbSubscription

Referenced by	Primary Key	Foreign Key
EcNbSubMatchExp_XREF	subscriptionId	subscriptionId
EcNbActionDefinition	subscriptionId	subscriptionId
EcNbActionQueue	subscriptionId	subscriptionId

Table 3-48. Dependencies on Table: EcNbActionDefinition

Referenced by	Primary Key	Foreign Key
EcNbOrderAction	actionId	actionId
EcNbNotificationAction	actionId	actionId
EcNbDpAction	actionId	actionId

Table 3-49. Dependencies on Table: EcNbActionQueue

Referenced by	Primary Key	Foreign Key
EcNbActionQueueLog	qPriority ActionId	qPriority ActionId

Table 3-50. Dependencies on Table: EcNbQueueInfo

Referenced by	Primary Key	Foreign Key
EcNbSubEventQueueRear	qPriority	qPriority
EcNbSubEventQueueFront	qPriority	qPriority
EcNbActionQueueRear	qPriority	qPriority
EcNbActionQueueFront	qPriority	qPriority
EcNbActionQueue	qPriority	qPriority
EcNbSubscribedEventQueue	qPriority	qPriority

Table 3-51. Dependencies on Table: EcNbSubscribedEventQueue

Referenced by	Primary Key	Foreign Key
EcNbSubEventQueueLog	qPriority eventId	qPriority eventId
EcNbEventMetadataDate	qPriority eventId	qPriority eventId
EcNbEventMetadataFloat	qPriority eventId	qPriority eventId
EcNbEventMetadataString	qPriority eventId	qPriority eventId
EcNbEventMetadataInteger	qPriority eventId	qPriority eventId
EcNbEventMetadataNose	qPriority eventId	qPriority eventId
EcNbActionQueue	qPriority eventId	Act_qPriority eventId

Table 3-52. Dependencies on Table: EcNbEventMetadataAttrDef

Referenced by	Primary Key	Foreign Key
EcNbEventMetadataDate	attributeName attributeType	attributeName attributeType
EcNbEventMetadataFloat	attributeName attributeType	attributeName attributeType
EcNbEventMetadataInteger	attributeName attributeType	attributeName attributeType
EcNbEventMetadataString	attributeName attributeType	attributeName attributeType
EcNbEventMetadataNose	attributeName attributeType	attributeName attributeType
EcNbMatchingExpression	attributeName attributeType	attributeName attributeType
EcNbEventAttrXref	attributeName attributeType	attributeName attributeType

Table 3-53. Dependencies on Table: EcNbEventDefinition

Referenced by	Primary Key	Foreign Key
EcNbSubscribedEventQueue	EventType ESDT_Id VersionID	EventType ESDT_Id VersionID
EcNbEventAttrXref	EventType ESDT_Id VersionID	EventType ESDT_Id VersionID
EcNbSubscription	EventType ESDT_Id VersionID	EventType ESDT_Id VersionID

3.1.9 Triggers

Sybase supports the enforcement of business policy via the use of triggers. A trigger is best defined as set of activities or checks that should be performed automatically by Sybase whenever a row is inserted, updated, or deleted from a given table. Sybase allows the definition of insert, update, and delete triggers for each table. A description of each the triggers in the SSS database is given in Table 3-54.

Table 3-54. Trigger Listing

Table Code	Trigger Name	Trigger Type
EcNbDpEventDetails	TrigInsEcNbDpEventDetails	Insert EcNbDpEventDetails: ensures that all data pool actions for an event are present in the table and then inserts them simultaneously into the data pool action queue.
EcNbEventDefinition	TrigInsEcNbEventDefinition	Insert EcNbEventDefinition: when a new ESDT is added, this trigger calls a stored procedure to get the subscribable attributes for the ESDT.

3.1.10 Stored Procedures

Sybase also includes support for business policy via the use of stored procedures. Stored procedures are typically used to capture a set of activities or checks that will be performed on the database repeatedly to enforce business policy and maintain data integrity. Stored procedures are parsed and compiled SQL code that reside in the database and may be called by name by an application, trigger or another stored procedure. A listing of each the stored procedures in the SSS database is given here. A brief definition of each of these stored procedures follows in Table 3-55.

Table 3-55. Procedure Listing (1 of 4)

Name	Description
ProcActionDequeue	Dequeues from the action queue.
ProcActionEnqueue	Enqueues into the action queue.
ProcActionReEnqueue	Re-enqueues the failed action.
ProcCancelSubscription	Cancels a subscription by setting its expiration date to “now” and its status to “canceled”.
ProcCheckDistributionEvent	Checks the existence of the distribution event.
ProcCheckExistingEvent	Checks the existence of the event.
ProcCleanupObsoleteESDT	Removes withdrawn event definitions marked for deletion, moving them to another table.
ProcDeleteProcessedEvent	Deletes information about an event.
ProcDeleteProcessedSub	Deletes information about a matched subscription.
ProcDeleteReEnqueue	ReEnqueues a failed deletion request.
ProcDeleteSubsByBO	Deletes all subscriptions associated with a given bundling order.
ProcDeleteSubsByTheme	Deletes all subscriptions associated with a given data pool theme.
ProcDeleteSubscription	Deletes a subscription.

Table 3-55. Procedure Listing (2 of 4)

Name	Description
ProcDequeueDeleteRequest	Dequeues from the list of deletables.
ProcGetAcquireInfo	Retrieves acquire infomation from tables.
ProcGetActionFilter	Returns a subset of actions in the action queue based on filtering criteria.
ProcGetActionInfo	Retrieves action infomation from tables.
ProcGetAttributeType	Retrieves attribute type from tables.
ProcGetAttrsForEventDef	Gets the metadata attributes that pertain to an ESDT.
ProcGetBOforSub	Gets the bundling order associated with a subscription.
ProcGetDataPoolActionInfo	Retrieves data pool action information.
ProcGetDeleteRequest	Gets delete request flag.
ProcGetDistributionInformation	Gets acquire information associated with a subscription.
ProcGetEmailInfo	Retrieves email infomation from database.
ProcGetEventCount	Returns the event count based on enqueueDateTime.
ProcGetEventMetadataFloat	Retrieves eventValueFloat from EventMetadataFloat table.
ProcGetExpiredSub	Gets expired subscription to delete.
ProcGetFailedActionInfo	Gets failed action info from action queue.
ProcGetGPolygonInfo	Retrieves gpolygon coordinates from sdsrv database.
ProcGetMatchingExpressionId	Retrieves MatchingExpressionId from sss database.
ProcGetMatchingSubscriptions	Identify matching subscriptions for an event.
ProcGetMetadata	Retrieves Metadata from ScienceDataServer database.
ProcGetNextActionId	Gets the next position in the action queue.
ProcGetNextMatchExpId	Gets the next position for matching expression.
ProcGetNextSubId	Gets the next id for the subscription.
ProcGetNoseMatchingId	Retrieves NoseMatchingExpressionId from sss database.
ProcGetOrbitInfo	Retrieves Orbit spatial Infomation from sdsrv database.
ProcGetQueueDiff	Returns the number of events and actions left to be dequeued from the runtime queues.
ProcGetRecoverAction	Gets failed action to recover
ProcGetRecoverEvent	Gets failed event to recover
ProcGetSensorInfo1	Retrieves Sensor Infomation from sdsrv database
ProcGetSensorInfo2	Retrieves Sensor Infomation from sdsrv database
ProcGetSensorInfo3	Retrieves Sensor Infomation from sdsrv database
ProcGetSpatialSearchType	Retrieves dbID, SpatialSearchType from sdsrv database
ProcGetSpatialType	Retrieves spatial search type from sdsrv database
ProcGetSubFilter	Gets subset of subscriptions.
ProcGetSubsForBundlingOrder	Gets the subscriptions associated with a particular bundling order.
ProcGetSubsForTheme	Gets the subscriptions associated with a particular data pool theme.
ProcGetThemeForSub	Gets the theme associated with a subscription.
ProcGetUserForSub	Gets the user associated with a subscription.
ProcInsEcNbDeleteRequestQueue	Inserts a row into the EcNbDeleteRequestQueue.

Table 3-55. Procedure Listing (3 of 4)

Name	Description
ProcInsEcNbEventMetadataNose	Inserts a row into the EcNbEventMetadataNose.
ProcInsEcNbEventTruth	Insert a row into the EcNbEventTruth.
ProcInsEcNbSubEventQueueLog	Insert a row into the EcNbSubEventQueueLog.
ProcInsertDateMetadata	Saves metadata about dates in the NBSRV database.
ProcInsertDistributionAction	Inserts a row into table EcNbDistribution. A duplicate key exception will be raised if the row already exists. The exception is handled by the action driver.
ProcInsertDpAction	Inserts a row into EcNbActionDefinition and EcNbDpAction.
ProcInsertDpEventDetails	Inserts a row into EcNbDpEventDetails.
ProcInsertFloatMetadata	Saves metadata about float values in the NBSRV database.
ProcInsertIntegerMetadata	Saves metadata about integer values in the NBSRV database.
ProcInsertMEDate	Saves date matching expr info in the NBSRV database.
ProcInsertMEFloat	Saves float matching expr info in the NBSRV database.
ProcInsertMEInteger	Saves matching expression info about int qualifiers.
ProcInsertMEString	Saves matching expression info about string qualifiers.
ProcInsertMetadata	Saves metadata about date, integer, float, string values in the NBSRV database.
ProcInsertNotificationAction	Inserts information about email action.
ProcInsertOrderAction	Insert a row into EcNbActionDefinition and EcNbOrderAction for an action.
ProcInsertStringMetadata	Saves metadata about string values in the NBSRV DB.
ProcInsertSubscription	Insert a new subscription.
ProcInsertTruth	Stores info about expressions that match metadata.
ProcQueueFailedAction	Removes actions that have been tried and failed more than the configured maximum number of times.
ProcResumeSubscription	Resume a particular subscription, i.e. change its status from Inactive to Active.
ProcResumeSubsByTheme	Resume all of the subscriptions associated with a particular data pool theme.
ProcSubscribedEventDequeue	Dequeues from the queue of newly arrived events.
ProcSubscribedEventEnqueue	Inserts new event into subscribed event queue.
ProcSubscribedEventReEnqueue	Re-enqueues failed events into the event queue.
ProcSuspendSubscription	Suspend a particular subscription, i.e. change its status from Active to Inactive.
ProcSuspendSubsByTheme	Suspend all of the subscriptions associated with a particular data pool theme.
ProcSynchEventAttr	Performs a one-time synchronization of ESDTs with their associated metadata attributes.
ProcSynchronizeESDTs	Synchronizes ESDT info with the SDSRV database.

Table 3-55. Procedure Listing (4 of 4)

Name	Description
ProcThemeHasSubs	Returns TRUE if there is at least one subscription associated with a particular theme; returns FALSE otherwise.
ProcUpdSubAddBundOrder	Updates a subscription with bundling order information.
ProcUpdSubAddTheme	Updates a subscription with theme information.
ProcUpdSubRmBundOrder	Updates a subscription by removing its association with a bundling order.
ProcUpdSubRmTheme	Updates a subscription by removing its association with a theme.
ProcUpdSubUserId	Updates a subscription by changing its userId.
ProcValidateDateTime	Used to validate a datetime string. If the string does not translate into a valid Sybase datetime value, an exception is raised.

3.2 File Usage

There are cases when the implementation of a persistent data requirement is better suited to a flat file than to a database table. A typical example of such data is system configuration information. System configuration information is fairly static and usually has no explicit relationship to other data in the enterprise. Another common use of files in ECS is as an interface mechanism between ECS and the external world. Files utilized in SSS are described herein.

3.2.1 Files Definitions

Not Applicable

3.2.2 Attributes

Not Applicable

3.2.3 Attribute Domains

Not Applicable

This page intentionally left blank.

4. Performance and Tuning Factors

4.1 Indexes

An index provides a means of locating a row in a database table based on the value of a specific column(s), without having to scan all data in the table. When properly implemented, indexes can significantly decrease the time it takes to retrieve data, thereby increasing performance. Sybase allows the definition of two types of indexes, clustered and non-clustered.

In a clustered index, the rows in a database table are physically stored in sequence-determined by the index. Clustered indexes are particularly useful, when the data is frequently retrieved in sequential order. Only one clustered index may be defined per table.

Non-clustered indexes differ from their clustered counterpart, in that, data is not physically stored in sorted order—newly added rows are stored at the end of the related database table.

A key of the types of indexes found in SSS is provided in Table 4-1 Index Type Key. A list and description of each of the defined indexes is given in Table 4-2 Index List.

Table 4-1. Index Type Key

Index Type Key	Description
PK	Primary Key
FK	Foreign Key
U	Unique – Only one for the column code combination
C	Clustered or non-clustered index
Sort	ASC (ascending) or DESC (descending) order

Table 4-2. Index List (1 of 3)

Table Code	Index Code	PK	FK	U	C
EcDbDatabaseVersions	EcDbDatabaseVersionsPkConstr	Yes	No	Yes	Yes
EcNbActionDefinition	PK_ACTIONDEFINITION	Yes	No	Yes	Yes
EcNbActionDefinition	EcNbActionDefSubIdIdx	No	No	No	Yes
EcNbActionDefinition	EcNbActionDefActTypeIdx	No	No	No	Yes
EcNbActionQueue	PK_ACTIONQUEUE	Yes	No	Yes	Yes

Table 4-2. Index List (2 of 3)

Table Code	Index Code	PK	FK	U	C
EcNbActionQueue	IX_actionQueueId_ActionQueue	No	No	No	Yes
EcNbActionQueue	EcNbActionQueueEventSubIdx	No	No	No	Yes
EcNbActionQueueFront	PK_ACTIONQUEUEFRONT	Yes	No	Yes	Yes
EcNbActionQueueLog	EcNbActionQLogDateTimeIdx	No	No	No	No
EcNbActionQueueLog	EcNbActionQLogPidIdx	No	No	No	No
EcNbActionQueueLog	PK_ACTIONQUEUELOG	Yes	No	Yes	Yes
EcNbActionQueueRear	PK_ACTIONQUEUEREAR	Yes	No	Yes	Yes
EcNbDeleteRequestQueue	EcNbDeleteRequestIdIdx	No	No	Yes	Yes
EcNbDistribution	PK_DISTRIBUTION	Yes	No	Yes	Yes
EcNbDpAction	PK_DPACTION	Yes	No	Yes	Yes
EcNbDpEventDetails	EcNbDpEventDetailsDbIdIdx	No			
EcNbDpEventDetails	PK_DPEVENTDETAILS	Yes	No	Yes	Yes
EcNbEventAttrXref	PK_EVENTATTRXREF	Yes	No	Yes	Yes
EcNbEventAttrXref	EcNbEventAttrXrefAttrIdx	No	No	No	Yes
EcNbEventDefinition	PK_EVENTDEFINITION	Yes	No	Yes	Yes
EcNbEventDefinitionObsolete	PK_EVENTDEFINITIONOBS	Yes	No	Yes	Yes
EcNbEventMetadataAttrDef	PK_EVENTMETADATAATTRIBUTEDEF	Yes	No	Yes	Yes
EcNbEventMetadataAttrDef	EcNbEventMetadataAttrTypeIdx	No	No	No	Yes
EcNbEventMetadataAttrDef	EcNbEventMetadataAttrFlagIdx	No	No	No	Yes
EcNbEventMetadataAttrDef	EcNbEventMetadataAttrParmIdx	No	No	No	Yes
EcNbEventMetadataDate	PK_EVENTMETADATADATE	Yes	No	Yes	Yes
EcNbEventMetadataFloat	EcNbMetaFloatPriorEventIdx	No	No	No	Yes
EcNbEventMetadataFloat	EcNbMetaFloatNameTypeIdx	No	No	No	Yes
EcNbEventMetadataInteger	PK_EVENTMETADATA_INTEGER	Yes	No	Yes	Yes
EcNbEventMetadataNose	PK_EVENTMETADATANOSE	Yes	No	Yes	Yes
EcNbEventMetadataString	PK_EVENTMETADATASTRING	Yes	No	Yes	Yes
EcNbEventMetadataString	IX_EventMetadataString_Event	No	No	No	Yes
EcNbEventMetadataString	IX_EventMetadataString_Attr	No	No	No	Yes
EcNbEventTruth	PK_EVENTTRUTH	Yes	No	Yes	Yes
EcNbEventTruth	EcNbEventTruthEventIdIdx	No	No	No	Yes
EcNbEventTruth	EcNbEventTruthMEIdIdx	No	No	No	Yes
EcNbMatchingExpression	PK_MATCHINGEXPRESSION	Yes	No	Yes	Yes
EcNbNoseMatchingExpression	PK_NOSEMATCHINGEXPRESSION	Yes	No	Yes	Yes
EcNbNotificationAction	PK_NOTIFICATIONACTION	Yes	No	No	Yes
EcNbOrderAction	PK_ORDERACTION	Yes	No	Yes	Yes
EcNbQueueInfo	PK_QUEUEINFO	Yes	No	Yes	Yes
EcNbSpatialMatchingExpression	PK_SPATIALMATCHINGEXPRESSION	Yes	No	Yes	Yes
EcNbSpatialMatchingExpression	xSpatialMatchingExpression	No	No	Yes	Yes
EcNbSubEventQueueFront	PK_SUBSCRIBEDEVENTQUEUEFRONT	Yes	No	Yes	Yes

Table 4-2. Index List (3 of 3)

Table Code	Index Code	PK	FK	U	C
EcNbSubEventQueueLog	EcNbSubEventQLogDateTimeIdx	No	No	No	No
EcNbSubEventQueueLog	EcNbSubEventQLogPidIdx	No	No	No	No
EcNbSubEventQueueLog	PK_SUBEVENTQUEUELOG	Yes	No	Yes	Yes
EcNbSubEventQueueRear	PK_SUBSCRIBEDEVENTREAR	Yes	No	Yes	Yes
EcNbSubMatchExp_XREF	PK_SUBMATCHEXP_XREF	Yes	No	Yes	Yes
EcNbSubMatchExp_XREF	EcNbSubMatchXREFmeIdIdx	No	No	No	Yes
EcNbSubMatchingExpDate	PK_MATCHINGEXPRESSIONDATE	Yes	No	Yes	Yes
EcNbSubMatchingExpFloat	PK_MATCHINGEXPRESSIONFLOAT	Yes	No	Yes	Yes
EcNbSubMatchingExpInteger	PK_MATCHINGEXPRESSIONINTEGER	Yes	No	Yes	Yes
EcNbSubMatchingExpString	PK_MATCHINGEXPRESSIONSTRING	Yes	No	Yes	Yes
EcNbSubscribedEventQueue	PK_SUBSCRIBEDEVENTQUEUE	Yes	No	Yes	Yes
EcNbSubscribedEventQueue	IX_eventQueueId_SubEventQueue	No	No	No	Yes
EcNbSubscribedEventQueue	EcNbSubscribedEventDataTypeIdx	No	No	Yes	Yes
EcNbSubscribedEventQueue	EcNbSubscribedEventGranuleIdx	No	No	No	Yes
EcNbSubscription	PK_SUBSCRIPTION	Yes	No	Yes	Yes
EcNbSubscription	IX_ESDT_Id_Subscription	No	No	No	Yes
EcNbSubscription	IX_VersionID_Subscription	No	No	No	Yes
EcNbSubscription	IX_EventType_Subscription	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionBundOrderIdx	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionThemeIdx	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionUserIdIdx	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionNumMEsIdx	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionStatusIdx	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionStartDateIdx	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionExpDateIdx	No	No	No	Yes
EcNbSubscription	EcNbSubscriptionLogicalIdx	No	No	No	Yes

4.2 Segments

Sybase supports the declaration of segments. A segment is a named pointer to a storage device(s). Segments are used to physically allocate a database object to a particular storage device. Segments defined for the SSS and all other subsystem databases are described in Table 4-3.

Table 4-3. Segment Descriptions

Segment Name	Description
Default	Default data segment used if no other segment specified in the create statement.
Logsegment	SYSLOGS, Transaction Logs
Systemsegment	System tables and indexes.

4.3 Caches

A cache is a block of memory that is used by Sybase to retain and manage pages that are currently being processed. By default, each database contains three caches:

Data cache – retains most recently accessed data and index pages

Procedure cache – retains most recently accessed stored procedure pages

User transaction log cache – transaction log pages that have not yet been written to disk for each user

The size of each of these default caches is a confugrable item which must be managed on a per DAAC basis. These caches may be increased or decreased by the DAAC DBA as needed.

The data cache can be further subdivided into named caches. A named cache is a block of memory that is named and used by the DBMS to store data pages for select tables and/or indexes. Assigning a database table to named cache causes accessed pages to be loaded into memory and retained. The named cache does not need to be allocated to accommodate the entire database table since the DBMS manages the cache according to use. Named caches greatly increase performance by eliminating the time associated for disk input and output (I/O). There are no named caches that are currently defined for the SSS Subsystem database. Named caches may be defined as the memory usage of the SSS database becomes more well known and the DAACs move into an operational environment. As named caches are defined this portion of the document will be updated.

5. Database Security

5.1 Approach

The database security discussed within this section is bounded to security implementation within the Sybase SQL Server DBMS. A Sybase general approach to security is adopted as illustrated in Figure 5-1.

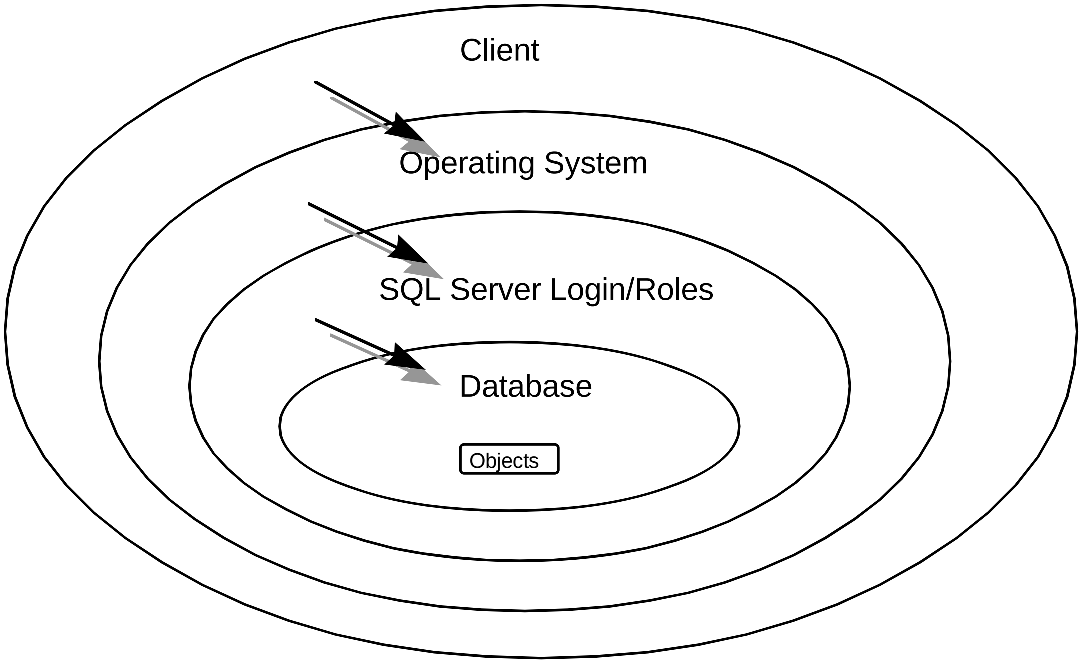

Figure 5-1. Sybase General Approach to SQL Server Security[1]

5.2 Users

The client (user) requires a SQL Server login to access the DBMS. The login is assigned to a user with certain related permissions for gaining access to particular objects (e.g., database tables, views, commands) within the database. The System Administrator may grant or revoke objects permissions for a login individually or based on defined group or roles.

[1] Reference Sybase Student Guide: *Advanced SQL Server Administration*.

5.3 Groups

Groups are a means of logically associating users with similar data access needs. Once a group has been defined, object and command permissions can be granted to that group. A user who is member of a group inherits all of the permissions granted to that group. The DAACs should define database groups to support the database security requirements of their individual DAACs. Assigning each user to the appropriate group should control security for local DAAC users.

5.4 Roles

Roles were introduced in Sybase to allow a structured means for granting users the permissions needed to perform standard database administration activities and also provide a means for easily identifying such users. There are six pre-defined roles that may be assigned to a user. A definition of each of these roles follows, as well as a description of the types of activities that may be performed by each role.

System Administrator (*sa_role*): This role is used to grant a specific user permissions needed to perform standard system administrator duties including:

- installing SQL server and specific SQL server modules
- managing the allocation of physical storage
- tuning configuration parameters
- creating databases

Site Security Officer (*sso_role*): This role is used to grant a specific user the permissions needed to maintain SQL server security including:

- adding server logins
- administrating passwords
- managing the audit system
- granting users all roles except the *sa_role*

Operator (*oper_role*): This role is used to grant a specific user the permissions needed to perform standard functions for the database including:

- dumping transactions and databases
- loading transactions and databases

Navigator (*navigator_role*): This role is used to grant a specific user the permissions needed to manage the navigation server.

Replication (*replication_role*): This role is used to grant a specific user the permissions needed to manage the replication server.

Sybase Technical Support (sybase_ts_role): This role is used to grant a specific user the permissions needed to execute database consistency checker (dbcc), a Sybase supplied utility supporting commands that are normally outside of the realm of routine system administrator activities.

The DAACs should review these roles and assign them to the appropriate login and/or groups.

5.5 Login/Group Object Permissions

During initial database installation logins used by the ECS custom code were created and permissions assigned for access to the SSS Subsystem database. In addition, special database installation login, ingest_role, was created to support database installation needs. For each login, the level of access is limited to that associated with their login, group or assigned group/role. Object Permissions are set within the installation scripts of the SSS Subsystem for each object and group/role.

Permissions are identified in Table 5-1. A specification of the object permissions is contained in Table 5-2.

Table 5-1. Permission Key

Permission	Description
A	All
S	Select
I	Insert
U	Update
D	Delete
E	Execute

Table 5-2. Object Permissions

Group	SYBASE LOGIN	Object	Permissions Granted					
			A	S	I	U	D	E
EcNbGroup	EcDsScienceDataServer EcNbActionDriver EcNbAlternateEventDriver EcNbDeleteRequestDriver EcNbRecoverDriver EcNbSubscribedEventDriver EcNbSubscriptionCLI EcNbSubscriptionGUI	All	X					

This page intentionally left blank.

6. Scripts

6.1 Installation Scripts

Scripts used to support installation of the SSS Subsystem database are listed in Table 6-1.

Table 6-1. Installation Scripts

Script File	Description
EcNbDbBuild	Create a new intialized SSS database.
EcNbDbPatch	Upgrade an existing SSS database to the next valid database version level.
EcNbInit	Initials specific tables in the SSS database.

6.2 De-Installation Scripts

No scripts used to support de-installation of the SSS Subsystem database.

6.3 Backup and Recovery Scripts

Scripts developed to perform backup and recovery of the SSS Subsystem database are listed in Table 6-2. These scripts should be configured to run automatically using the Unix chron facility. It is recommended that, transaction logs dumps (incremental dumps) are performed a minimum of 3 times each day. It is recommneded that database dumps (full database dumps) are performed a minimum of once each day. Backup and recovery are M&O activities. At their discretion, DAACs may modify these backup/recovery scripts or utilize backup/recovery scripts developed by their local M&O staff.

Table 6-2. Backup and Recovery Scripts

Script File	Description
EcCoDbSyb_DumpDb	Dumps all databases for managed by the SQL server instance.
EcCoDbSyb_DumpTran	Dumps the transaction log for all databases managed by the SQL server.

6.4 Miscellaneous Scripts

Miscellaneous scripts applicable to the SSS Subsystem database are listed in Table 6-3.

Table 6-3. Miscellaneous Scripts and Input Data Files

Script	Description
EcCoDbSyb_CkErrorLog	Checks the error log for error messages warranting DBO attention. Superseded by DbVision.
EcCoDbSyb_DbStat	Updates index statistics for each table in the selected database.
EcCoDbSyb_DboMail	Emails DBA error notification via e-mail. Used by EcCoDbSyb_DumpDb/Tran and EcCoDbSyb_CkErrorLog scripts.

Appendix A. Entity Relationship Diagram

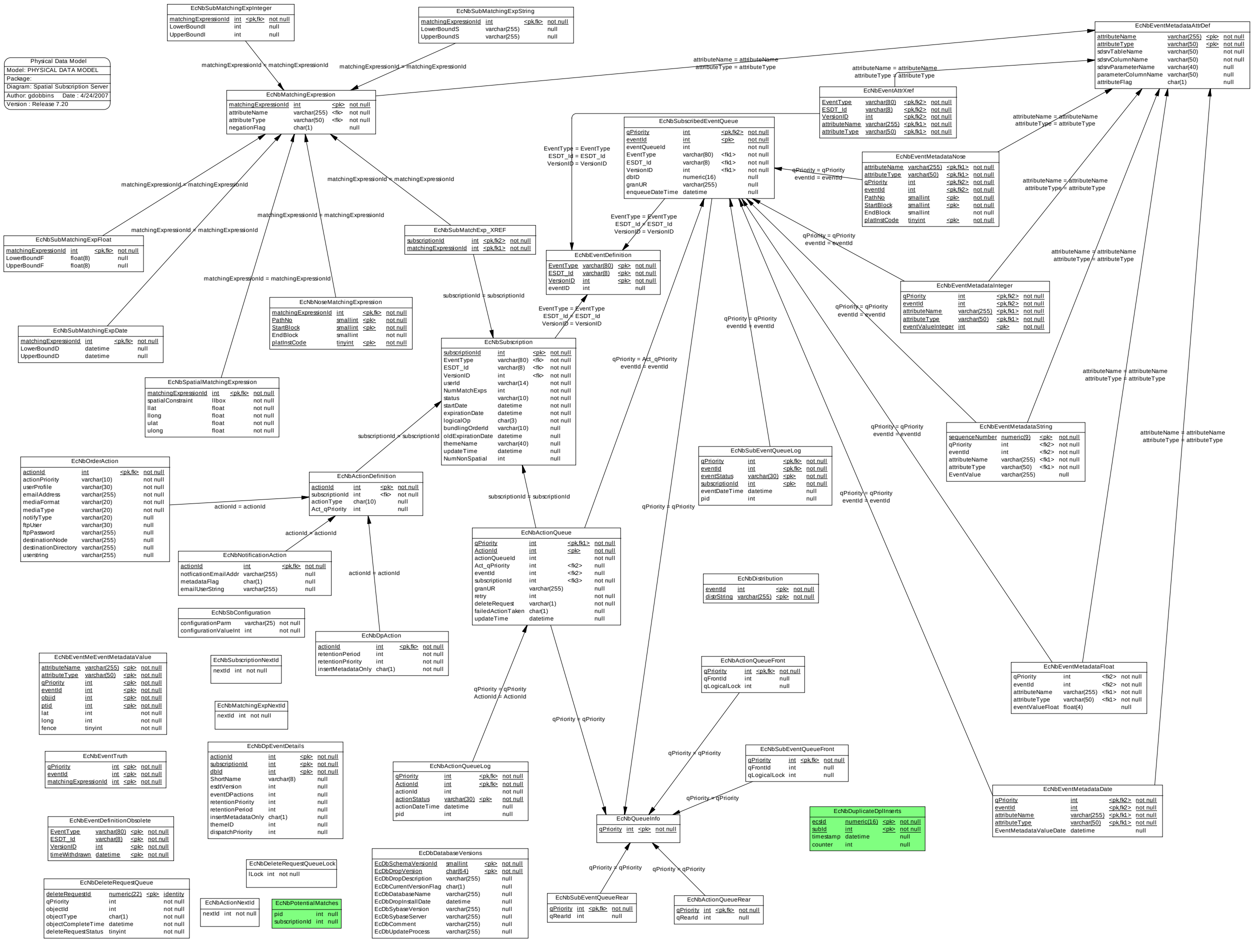

Figure A-1. Spatial Subscription Server

This page intentionally left blank.

Abbreviations and Acronyms

ADSRV	Advertising Service CSCI
ANSI	American National Standards Institute
CASE	Computer Aided Software Engineering
CD	contractual delivery 214-001
CDRL	contract data requirements list
CDS	cell directory service
CI	configuration item
COTS	commercial off-the-shelf (hardware or software)
CSCI	computer software configuration item
CSDT	Computer Science Data Type
CSMS	Communications and Systems Management Segment (ECS)
CSS	Communications Subsystem
DAAC	Distributed Active Archive Center
DBMS	Database Management System
DCN	Document Change Notice
DDICT	Data Dictionary CSCI
DDIST	Data Distribution Services CSCI
DDN	Data Delivery Notice
DID	data item description
DM	Data Management
DMS	Data Management Subsystem
DP	Data Provider
DPL	Data Pool Subsystem
DPS	Data Processing Subsystem
DSS	Data Server Subsystem
ECS	EOSDIS Core System
EDC	EROS Data Center
EDHS	ECS Data Handling System

EDOS	EOS Data and Operations System
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
ERD	Entity Relationship Diagram
EROS	Earth Resources Observation System
ESDIS	Earth Science Data and Information System (GSFC)
ESDT	Earth science data types
FK	Foreign Key
FTP	File Transfer Protocol
GSFC	Goddard Space Flight Center
GUI	graphic user interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
HWCI	Hardware Configuration Item
ICD	interface control document
ID	identification
INGEST	Ingest Services CSCI
I/O	Input and Output
IOS	Interoperability Subsystem
IP	Internet Protocol
ISS	Internetworking Subsystem
IV&V	independent verification and validation
LaRC	Langley Research Center (DAAC)
MCF	Metadata Configuration File
MSFC	Marshall Space Flight Center
MSS	Management Support Subsystem
NOSE	Nominal Orbit Spatial Extent
PDF	Portable Document Format
PDPS	Planning and Data Processing Subsystem
PK	Primary Key
PLANG	Production Planning CSCI

PLS	Planning Subsystem
RDBMS	Relational Data Base Management System
RPC	Remote Procedure Call
SDSRV	Science Data Server
SSS	Spatial Subscription Server
STMGT	Storage Management Software CSCI
SUBSRV	Subscription Server
UR	Universal Reference
WWW	World-Wide Web

This page intentionally left blank.