1. Introduction and Documentation Index

The Service Negotiation Support (SNS) building block is a processing layer building block (PLBB) that handles the network-related negotiation functions. In accordance with OSCA™ architecture guidelines, these functions are defined in terms of *contracts* that specify how other systems (e.g., a service negotiation user layer building block, SN ULBB, or account-based negotiation system, ABNS) interact with SNS to request negotiation assistance. This document contains the specifications for the contracts that are available in the SNS Release 2.0. It first explains the purpose of each contract and its associated processing, and then describes the contract interfaces.

This document is organized into six sections and six appendixes.

Section 1 — "Introduction and Documentation Index"

Section 2 — "SNS Contracts" describes the structure of the SNS contract set.

Section 3 — "SNS Contracts: FCIF Messages" provides an overview of the FCIF message structure of the SNS contracts.

Section 4 — "Negotiation Stand-Alone Contracts" presents a detailed explanation of the sections in the negotiation stand-alone contracts.

Section 5 — "Negotiation Dialogue Contracts" provides a detailed discussion of the sections in the negotiation dialogue contracts.

Section 6 — "System Administration Table Maintenance Contracts" presents an examination of the sections in the table maintenance contracts.

Appendix A — "FCIF Overview" reviews the basic rules for structuring statements in the FCIF language.

Appendix B — "SNS FCIF Tag Descriptions" lists and describes all FCIF tags used by SNS.

Appendix C — "Rules for the ADDR Data Aggregate" explains the rules for the different types of input address.

Appendix D — "SNS Return Codes" provides information on the return codes that may appear in the response messages of SNS contracts.

OSCA is a trademark of Bellcore.

Appendix E – "SNS Contract Examples" shows examples of request and response messages.

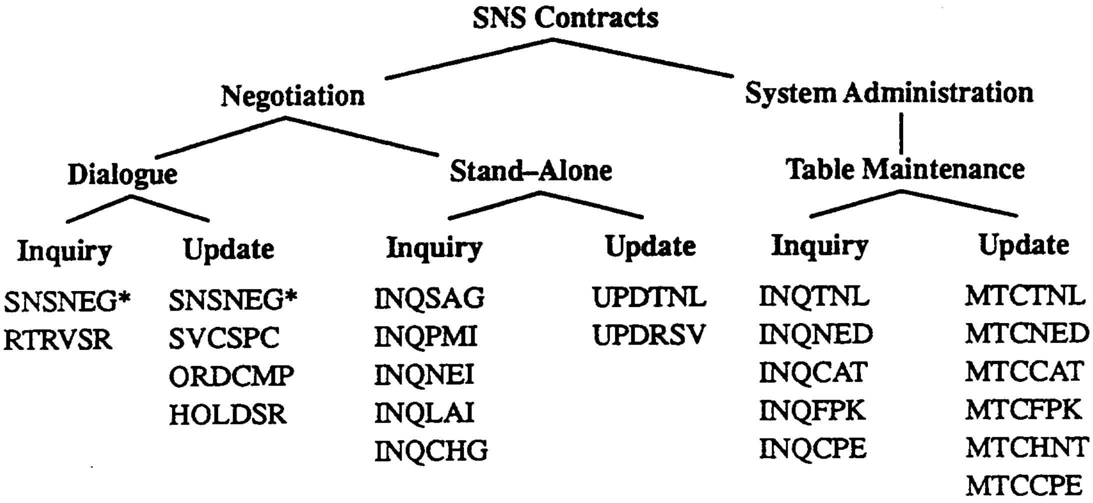
Appendix F – "Guide to Acronyms" lists and defines the major acronyms used in this document.

In addition to this *Service Negotiation Support (SNS) Contract Specifications Guide*, there are several other Bellcore documents that support SNS. These documents are:

1. *Service Negotiation Support (SNS) System Administration Guide*, BR 007-560-405, covers application installation, security, recovery, troubleshooting, and other topics.
2. *Service Negotiation Support (SNS) Database Administration Guide*, BR 007-560-406, describes the SNS databases and explains database installation, recovery, and maintenance.
3. *PREMIS/NMAG Application Guide*, BR 007-560-408, explains the procedures for working with the Maintenance SVA screen and for producing reports.
4. *PREMIS/NMAG Database Administration Guide*, BR 007-560-409, describes the PREMIS system marketing database area and provides the physical data specifications for the area and the records.

2. SNS Contracts

SNS is an OSCA processing layer building block (PLBB). In accordance with the OSCA architecture building block principles, interactions between SNS and other building blocks are defined by contracts. A diagram of the SNS 2.0 contract set is presented in Figure 2-1. This contract set is divided into two main groups: *negotiation* contracts and *system administration* contracts. The Negotiation contracts support the negotiation process. The System Administration contracts support database table maintenance.



*Can be either Inquiry or Update, depending on input section

Figure 2-1. SNS 2.0 Contract Set

2.1 Negotiation Contracts

Within the negotiation contract subset, a contract is classified as *stand-alone* or *dialogue*. To understand the difference between stand-alone contracts and dialogue contracts it is important to understand the concept of an SNS session. A session represents a set of activities that are performed to process one or more related contract invocations. Each session is uniquely identified by a session identifier that must appear in all incoming messages for the session. The absence of the session identifier triggers a new session to be *established* in SNS. When this happens, a session identifier is assigned. As SNS processes the contract, it retains important session data that tracks the history of events that have occurred in the session. If the session is *suspended*, SNS saves the session data so that it is available when the session is later *resumed*. Session data is eventually purged from SNS when the session is *terminated*.

A key difference between stand-alone and dialogue contracts is the session lifespan and when session termination occurs.

- When a stand-alone contract is invoked, SNS always establishes a new session to handle the request. SNS may suspend and resume the session when interactions with other systems (e.g., PREMIS and LFACS) are involved. However, the session is immediately terminated after the contract is processed and a response returned to the contract invoker. As a result, there is no history of the request in SNS once it is complete. A stand-alone contract, therefore, is always processed independently of other contracts.
- By contrast, SNS creates a new session for a dialogue contract only the first time that it is invoked to start a negotiation dialogue. The same session handles all subsequent dialogue contract invocations related to the negotiation. After SNS processes a dialogue contract, it may suspend the session, thereby allowing the dialogue to continue at some later point in time. Session data is therefore available throughout the dialogue to provide a history of the negotiation. This data is valuable in handling subsequent requests. SNS does not terminate the session until the negotiation is complete. When the session is terminated, the negotiation dialogue ends.

To summarize, the extended duration of the session allows dialogue contracts to be tied to a particular negotiation in SNS for the duration of that negotiation. Stand-alone contracts, on the other hand, are treated as independent and isolated requests because the session ceases to exist after the contract is processed.

2.1.1 Negotiation Contract Usage

Although stand-alone and dialogue contracts provide some overlapping functionality, they are designed to complement each other. Each contract is intended to be used at a different stage in the negotiation, under different circumstances. The following sections describe the stand-alone and dialogue contracts and place each in a probable negotiation stage. However, the sequence of contracts is flexible, and stand-alone and dialogue contracts can be combined in whatever order the contract invoker chooses.

2.1.1.1 Stand-Alone Negotiation Contracts

There are seven stand-alone contracts:

- Street Address Validation Inquiry (INQSAG)
- Network Element Inquiry (INQNEI)
- Product Marketing Inquiry (INQPMI)
- Change in Service Inquiry (INQCHG)
- Loop Availability Inquiry (INQLAI)
- Loop Reservation Update (UPDRSV)
- TN List Selection/Return (UPDTNL)

Certain stand-alone contracts are useful in the early fact finding stages of the negotiation when the negotiator narrows down what service/product the customer is interested in ordering. Stand-alone contracts support various activities that the negotiator may need to perform at this stage of the negotiation:

- Identify a customer's location and local serving office (INQSAG)
- Determine network elements (e.g., switches) in a customer's local serving office (INQNEI)
- Determine product availability at a customer's location (INQPMI)
- Determine loop availability at a customer's location (INQLAI)
- Determine whether a customer's working service can be regraded using existing outside plant facilities (INQCHG).

Following the preliminary fact finding, the actual negotiation of the service takes place to gather the data needed to order the requested service/product. At this time, the customer may need new telephone numbers (TNs). The UPDTNL contract is available to select TNs from the local TNLIST database in SNS. UPDTNL can also be used to return unused TNs, so that valuable TN resources are not wasted. However, this responsibility is left to the TN selector.

After the customer has placed an order, the UPDRSV contract may be invoked to perform a loop reservation. Should the customer cancel the order, the loop reservation can be removed using the same contract. UPDRSV is also used to modify a loop reservation, in the event the customer requests a change.

2.1.1.2 Negotiation Dialogue

The goal of the negotiation dialogue in SNS is to provide the negotiation-based data needed by the provisioning processes to produce a service request. Dialogue contracts are used to negotiate the specific details of the customer's service, based on network availability. SNS supports negotiations of locally switched Digital Centrex services provided from the AT&T 5ESS® and Northern Telecom DMS®-100 switches. Dialogue contracts also support automated network resource management. This added capability alleviates the need for BCCs to engineer and build this logic into their account-based negotiation systems or service negotiation user layer building blocks (ULBBs).

2.1.1.2.1 Line Negotiation Support in SNS

During a negotiation dialogue, SNS provides network data and resources to assist in the negotiation of Digital Centrex lines and groups. All interactions with SNS to obtain this data are performed on a *service request* basis. A service request corresponds to one of the following negotiation activities:

- *Add line(s)* to an existing Digital Centrex service
- *Change* an existing customer's Digital Centrex service.

At the time the service request is established in SNS, critical service request data must also be provided so that SNS can determine what network resources are involved. The negotiation activity dictates what kind of additional data is needed. All subsequent network-related activities performed in SNS for this service request rely on this data. A service request modification must be made in SNS whenever a change in service request data occurs, in case different network resources are involved.

A new service request triggers certain preliminary activities that set the stage for the remainder of the negotiation. First, SNS assigns a Service Request Number (SRN) to identify the service request. Subsequent processing depends on the type of negotiation activity being performed. Network element determination is an important step in the creation of a new service request. For a Digital Centrex

5ESS is a registered trademark of AT&T.
DMS is a registered trademark of Northern Telecom Inc.

customer, SNS determines the serving network element (e.g., switch) based on the customer's main Centrex TN. The customer's serving network element influences network availability and network resource selection.

During the negotiation of the service request, the negotiator may need access to network data. SNS supports various network data inquiries to assist the negotiator in determining network availability and compatibility. SNS also supports an additional inquiry that provides the in-service view of a series completion hunt group.

As a provider of network data during the negotiation process, SNS not only supplies this data when requested, but also determines whether previously requested data is still usable. Whenever a modification is made to the original service request data, SNS presents a warning if network data provided earlier may no longer be pertinent. Alerting the invoking building block when these situations occur assists in "salvaging" as much of the negotiation as possible without jeopardizing the integrity of the data being used. It also reduces the likelihood that inaccurate information will be used during the negotiation.

SNS selects and manages TN resources on behalf of the negotiator. In a negotiation dialogue, TNs are specifically chosen for the service request and are not meant to be used in any other negotiation. Enhanced management of TN resources, not provided by stand-alone contracts, is also supported. SNS keeps track of selected TNs for this purpose. If the service request data is later modified, SNS can determine whether the TNs already selected are still capable of being used. SNS has the knowledge to make this decision, based on its internal selection algorithms which rely on the session history. This ensures that the appropriate resources will be used on the eventual service specification and included on the ultimate service order.

The negotiator may *cancel* the service request in SNS, provided service orders have not yet been issued. Upon cancellation, SNS returns all TNs selected for the service request. The service request is then removed from SNS. If service orders have already been issued, the service request cannot be canceled. In this situation, SNS waits for the cancellation pass of the order instead. (This is discussed in greater detail in the "Order Completion and Cancellation Activities in SNS" section below.) Eventually, when SNS receives the order cancellation, it will perform the necessary clean-up activities.

2.1.1.2.2 Service Specification Activities in SNS

Once the Centrex line negotiation is complete and the requested service has been confirmed with the customer, a service specification is passed to SNS. In SNS 2.0, only a minimal set of service specification data is needed to identify any service orders that were issued in the negotiation. It is important to establish the relationship between service order and negotiation dialogue, so that the completed order(s) can be related back to the appropriate session.

2.1.1.2.3 Order Completion and Cancellation Activities in SNS

SNS is notified of post-completion and cancellation orders so that it can maintain the TNLIST database and clean up negotiation dialogues that have completed. On a successful order completion, SNS updates the in-service views of series completion hunt groups stored in the TNLIST database. If all orders associated with the negotiation dialogue have been completed (or were canceled), SNS returns any unused network resources. The session and its related data are also purged. This concludes the negotiation dialogue.

2.1.1.2.4 Session Support in SNS

A negotiation dialogue may involve multiple contacts with the customer over an extended period of time. The negotiator may interrupt the negotiation dialogue and *hold* the service request data in SNS so that it can be *retrieved* at a future point (e.g., to complete the negotiation or correct the negotiation data). This option is available to those building blocks (e.g., service negotiation ULBB) that might not have a permanent data store facility. SNS serves as a data repository by holding the negotiation data. When negotiation of the service request is resumed, this data can be retrieved from SNS to restore the environment. The negotiator can then continue negotiating the service request or correct/cancel any issued orders.

2.1.1.2.5 Dialogue Contracts

Five contracts support the negotiation dialogue:

1. The *Line Negotiation Support* contract (SNSNEG) is a multi-purpose contract that assists in negotiating Digital Centrex lines and groups. This contract provides the following capabilities.

   - Service Request Administration: Establish, modify or cancel a service request.
   - Network Data Inquiries: Assist the negotiator in determining network availability and compatibility for a service request, and provide in-service views of series completion hunt groups in the TNLIST database.
   - Network Resource Selection: Select/return TNs for a service request.

   The *Line Negotiation Support* contract is designed so that the Bellcore Client Companies (BCCs) can pick and choose which request(s) to use in their negotiation flows when it is appropriate to do so.

2. An abridged version of the *Service Specification* contract (SVCSPC) is provided in SNS 2.0 to identify the associated service orders issued during the negotiation dialogue.
3. The *Hold* contract (HOLDSR) supports the capability to put the negotiation dialogue on hold and save the service request data in SNS. This data can be used later to restore the session when the negotiation dialogue is resumed.
4. The *Retrieve* contract (RTRVSR) supports the capability to retrieve the service request data previously held in SNS (via the *Hold* contract) and to take the negotiation dialogue off hold.
5. In addition to these four contracts that are used during the negotiation, SNS also supports an additional *Order Completion* contract (ORDCMP) that follows the negotiation and effectively ends the negotiation dialogue. This contract, which is invoked on an order basis, signifies the completion or cancellation of an order. As orders associated with a negotiation dialogue complete, SNS updates the TNLIST database in-service views. If all orders have completed, SNS also returns all unused network resources selected for this negotiation. The session and its related data are purged, thereby ending the negotiation dialogue.

2.2 System Administration Contracts

Within the system administration contract subset, all contracts are classified as *Table Maintenance*. Table Maintenance contracts can be one of two types, inquiry or update. The difference between inquiry contracts and update contracts is that only update contracts alter the contents of the database tables. An inquiry contract accesses the database tables and returns the data being requested to the contract invoker. An update contract provides the means to alter the contents of database tables through build, remove, and change functions.

2.2.1 Table Maintenance Inquiry Contracts

There are five inquiry contracts:

1. Inquiry of TNLIST data (INQTNL)
2. Inquiry of Feature Group Catalog, Feature Catalog, and Name Description data (INQCAT)
3. Inquiry of Network Element and Switch Parameter data or Switch Feature Exception data (INQNED)

4. Inquiry of CPE data (INQCPE)
5. Inquiry of Feature Package data (INQFPK)

Inquiry contracts are useful to verify information already stored in the database tables and to retrieve stored data for use as models to create new table entries. Inquiry activities that the maintenance personnel may need to perform include, but are not limited to:

- Obtain a list of all telephone numbers within a Centrex Group status (INQTNL)
- Obtain a list of all features and related service information including description names for a specified switch type/generic (INQCAT)
- Obtain a list of attributes associated with a specific CPE (INQCPE)
- Obtain a list of all features under a specific feature package (INQFPK)
- Obtain a list of all feature exceptions at a switch (INQNED).

These inquiries may then be followed by additional maintenance activity to effect changes to the database tables.

2.2.2 Table Maintenance Update Contracts

The table maintenance update contracts support the build, remove, and change functions for the database tables. The update contracts can be used by maintenance personnel to perform the following operations:

- Build, remove, or change data associated with a TNLIST Group (MTCTNL)
- Build, remove, or change Series Completion Hunt Groups and their associated telephone numbers (MTCHNT)
- Change Feature Group Catalog data, Feature Catalog data, and Name Description data (MTCCAT)
- Build, remove, or change data associated with Customer Premises Equipment (MTCCPE)
- Build, remove, or change Feature Packages and their associated data (MTCFPK)
- Build, remove, or change Network Element data (MTCNED)
- Build, remove, or change Switch Feature Exception data (MTCNED).

Data stored in the database tables through table maintenance update contracts is used to aid the service negotiation process and to enable the negotiator to perform telephone number selection.

3. SNS Contracts: FCIF Messages

All SNS contracts are defined using the Flexible Computer Interface Form (FCIF) format. (See Appendix A for general specifications for the FCIF syntax.)

The contract data in an FCIF message is a string of ASCII characters. The FCIF sections and data aggregates logically group the data.

- An FCIF section is the first level of logical data groupings in the FCIF message. The allowable sections that may appear in each message depend upon the contract and the message type (request or response).
- Within the FCIF section, one or more FCIF tag/value pairs and/or data aggregates may appear.
  - A *tag* is the only item that may contain a data *value*.

    Although FCIF does not impose a restriction on the length of a tag name, SNS limits tag names to eight characters. Appendix B provides a description of the tags defined for the SNS 2.0 contracts.
  - A data *aggregate* is used to logically group data items within an FCIF section (or data aggregate in a nested situation).

Special FCIF delimiters (* { } % ; =) are used to delineate the start and end of FCIF sections, tags, values, and aggregates in the FCIF message.

Since a contract is defined in FCIF, it is understood that the messages adhere to the FCIF rules explained in Appendix A, unless otherwise stated. The contract definition specifies how a "valid" message should be constructed for a particular contract and thereby imposes an *additional* set of constraints on the message, over and above the standard FCIF rules.

3.1 SNS Contract Design

SNS contracts use common conventions to promote consistency across contracts. In general, SNS contracts consist of two types of FCIF messages:

- The first is a "Request" message from the contract invoker to SNS. This message *invokes* an SNS contract and contains the data needed to process that contract.
- The second is a related "Response" message from SNS back to the contract invoker or some other pre-determined destination. This message indicates the results of processing the contract.

Each request and response message is further divided into FCIF sections. The FCIF sections defined for a message can be categorized into the following two types:

- *Message header* sections contain environment-related data that is administrative in nature. When such a header appears in an FCIF message, its scope is the *entire* message.

  The same header sections are defined for all SNS contracts: control header, *PLHDR, *UMSG, and *SNSHDR sections. The control header section must appear first followed by the *PLHDR section. The remaining sections can appear in any order.

- *Contract data* sections contain the data related to the specific function(s) that the contract supports.

  - In a request message, a contract data section contains the data required to invoke the specific activity. Depending upon the contract, it may contain one or more requests.

  - In a response message, a contract data section contains the results of processing each request that appeared in the original request message.

  Multiple contract data sections are defined for some contracts.

3.1.1 Contract Input: Request Message

SNS request messages contain a combination of message header and contract data FCIF sections that serve as the contract input. Each contract interface specifies which FCIF sections may be included in the request message. The tags and data aggregates that comprise each section are also identified in the interface specification.

Contract data sections contain one or more requests that conform to the same basic structure. Typically, control information is provided in a CTL aggregate, which identifies the function to be performed. The ACL aggregate contains the action line data needed to process the request. At times, two ACL aggregates are needed so that both an *old* view and a *new* view of the data can be supplied. When more than one request is allowed in a section, the data for each request is grouped in a REC aggregate.

3.1.2 Contract Output: Response Message

After SNS processes a request, it may send a response message. This message provides important information about whether SNS was successful in performing the requested function(s).

Like the request message, the response message contains a combination of message header and contract data FCIF sections. Typically, SNS returns the original input request data, supplemented with additional return codes that identify the processing conditions that were encountered. In many cases, SNS also returns additional output data that was either specifically requested or that is useful in handling an error.

SNS uses various means of conveying information to the contract invoker to indicate whether it was successful in performing the requested activity. Depending upon what situations were encountered, SNS may provide information in various locations in the response message.

- Fatal error conditions are identified in the *UMSG section.
- In a negotiation dialogue, service request timing problems are identified in the *SNSHDR section. In this situation, SNS does *not* attempt to continue processing any of the negotiation requests in the message.
- Status conditions related to processing the request are returned in the appropriate contract data section.

3.1.2.1 SNS Return Codes

Return codes are placed in the response message to indicate whether SNS was successful in processing a contract. One or more return codes may be embedded within the *SNSHDR or contract data section to identify one or more processing conditions. They are provided so that the invoking building block can analyze what happened in SNS and perform the necessary actions.

Each return code consists of two elements:

- The *tag name* identifies the type of processing condition encountered. Return code tag names are limited to eight characters in length and always end with the characters "RC." For example, the IERC tag name represents an input error condition.
- The *value* conveys the meaning of the return code. The value consists of three numeric characters and an optional suffix, which is appended to the value (e.g., 102-1). The suffix is useful when there are variations in processing conditions.

The placement of the return code within the FCIF message must also be taken into consideration when analyzing the return code. Its meaning is dependent on the FCIF section in which it appears, since a return code may appear in different FCIF sections.

Return codes are *not* meant to function as user error codes. SNS, as a PLBB, makes no assumptions about how return code information is displayed to the end user (if at all). Such presentation issues are left strictly to the service negotiation ULBB.

3.2 FCIF Message Conventions Used in This Document

The FCIF messages shown in this document contain various data items – tags and data aggregates (logical groupings of data items) – that may appear in each SNS contract section. Tags and aggregates that exist at the same level in the FCIF hierarchy are not expected to be sequenced in any particular order, unless specifically stated. Similarly, FCIF sections are not guaranteed to appear in any particular order, unless specifically stated. In an actual FCIF message, the sequence of sections, tags, and aggregates may *not* match the order in which they are presented in this document.

The following conventions are used in the FCIF messages in this document:

- The left column contains the actual FCIF specification. It identifies the FCIF section, tag, or aggregate name and the accompanying FCIF delimiters. Note that, in the actual FCIF message, all SNS tag names are in *upper* case only.
- A data aggregate name is followed by an open curly bracket ({). A close curly bracket (}) will appear on a line by itself at the end of the aggregate.
- Tag names are followed by an equal sign, three dots, and a semicolon (=...;). In an actual FCIF message, a value would appear in place of the three dots. When a specific value is required for a tag, the value is shown in the left column. When there is a set of valid values for a tag, the valid code set is identified in angled brackets, e.g., <A,B>, in the right column. (See Appendix B, "SNS FCIF Tag Descriptions," for more information on tags.)
- Data items that belong to an aggregate are shown directly below the data aggregate and are indented. When the data items associated with an aggregate are not shown, a series of dots (...) is used to represent them.
- The right column contains the aggregate name or a text description of the tag.
- A tag is required, unless denoted optional or conditional. This information is provided in square brackets in the right column. The following table shows the optional and conditional notation for tags.

Table 3-1. FCIF Tag Notation

FCIF Notation	Description
[opt]	Tag is always optional.
[opt when ...]	Tag is an option only under certain circumstances. Otherwise, tag is not applicable.
[conditional - required when ...]	Tag is conditionally required under certain circumstances. Otherwise, tag is not applicable.

- The determination of whether an aggregate is required or optional depends on the data items that make up the aggregate. An aggregate that consists of all optional data items is itself considered optional, unless otherwise stated. On the other hand, an aggregate that consists of at least one required data item is considered required, *unless* specifically denoted optional. (The square bracket FCIF notation described above is also used for data aggregates.)
- When a tag or aggregate may appear multiple times, this is denoted by curly brackets, e.g., {1-N}.
- The FCIF messages shown in this document present each aggregate name and tag on a line by itself, with tags indented under the aggregate to which they belong. This format is provided for readability purposes only. Actual FCIF messages do not have the new-line and indented format.

The following sections describe the FCIF header sections that may be used by all SNS contracts. These sections are: Control, Platform Header, User Message, and SNS Header. Section 4 in this document explains the FCIF message sections that apply specifically to the negotiation stand-alone contracts. Section 5 presents the FCIF sections included in the negotiation dialogue contracts. Section 6 describes the FCIF sections that pertain to the system administration table maintenance contracts. (Sample FCIF messages can be found in Appendix E.)

3.3 FCIF Header Sections

This section explains the structure of the FCIF header sections that are defined in every SNS contract.

3.3.1 FCIF Control Section

All FCIF messages **must** begin with an FCIF Control Section (FCIF Header). This FCIF header is actually an unnamed FCIF section that consists of a single tag/value pair. In a request message, the name of the tag is C1. In a response message, the tag name is C2. Thus, the header is commonly referred to as the *C1 or *C2 header. The value associated with the tag has a fixed field format and a length of 64 characters. Not all fields defined in the header are used.

Position	**Data**
1-3	Tag
4-5	Delimiter (=)
6-11	Contract Name
12-27	Service Request Number (SRN)/Session ID
28-33	Unused
34-41	Origin
42-49	Destination
50-57	Unused
58	SNS Entity ID
59	Solicited/Unsolicited Indicator
60	Transaction Priority
61	Unused
62	Message Format Indicator
63-64	Terminators (;%)

The following fields in the header are used in each contract message:

- Tag — For messages coming into SNS, this field has a value of "*C1". For messages going back to the contract invoker, this field has a value of "*C2".
- Contract Name — This field identifies the contract being invoked in SNS.

  The contract names for the negotiation stand-alone contracts are:

Table 3-2. Negotiation Stand-Alone Contract Names

Contract Name	Description
INQSAG	Street Address Validation
INQPMI	Producting Marketing Inquiry
INQNEI	Network Element Inquiry
INQLAI	Loop Availability Inquiry
INQCHG	Change in Service Inquiry
UPDTNL	TN List Selection/Return
UPDRSV	Loop Reservation

The contract names for the negotiation dialogue contracts are:

Table 3-3. Negotiation Dialogue Contract Names

Contract Name	Description
SNSNEG	Line Negotiation Support
SVCSPC	Service Specification
HOLDSR	Hold
RTRVSR	Retrieve
ORDCMP	Order Completion

The contract names for the system administration table maintenance contracts are:

Table 3-4. Table Maintenance Contract Names

Contract Name	Description
INQTNL	TN Data Inquiry
INQNED	Switch-Related Data Inquiry
INQCAT	Catalog Data Inquiry
INQFPK	Feature Package Data Inquiry
INQCPE	CPE Data Inquiry
MTCTNL	TN Data Maintenance
MTCNED	Switch-Related Data Maintenance
MTCCAT	Catalog Data Maintenance
MTCFPK	Feature Package Data Maintenance
MTCHNT	Hunt Group Maintenance
MTCCPE	CPE Data Maintenance

- Service Request Number (SRN)/Session ID — This field contains the service request number for negotiation dialogue contracts and the session ID for negotiation stand-alone contracts. This field is not applicable to system administration table maintenance contracts and should be left blank.

  The unique *service request number (SRN)* identifies the service request. When the service request is initially established in SNS via the SNSNEG contract, this field should be left blank in the *C1 header. When SNS responds, it populates the *C2 header with the new SRN. If the service request was successfully established in SNS, this SRN should be supplied in the *C1 header of all subsequent SNSNEG, SVCSPC, HOLDSR, and RTRVSR contract invocations related to this service request. If the service request was *not* established in SNS, the SRN is returned for debugging purposes only.

  When the ORDCMP contract is invoked, the SRN may be left blank if unknown. In this case, SNS derives the SRN based on the order identifier. If the order is not associated with a negotiation dialogue in SNS, a new SRN is assigned.

  The *session ID* is the unique identifier of the negotiation stand-alone session. The contract invoker should *always* leave this field blank in the *C1 header. When SNS responds, it populates the *C2 header with the new session ID. This data is supplied for debugging purposes only.

- Origin – For messages coming into SNS, this field identifies the contract invoker. For messages going back to the contract invoker, this field has a value of "SNS."
- Destination – For messages coming into SNS, this field has a value of "SNS". For outgoing messages, this field identifies the destination to which the response message is to be routed. Response messages for all SNS contracts, except the ORDCMP contract, are returned to the contract originator. Consequently, this field contains the originator identified in the *C1 header of the request message. On the other hand, ORDCMP response messages are not routed back to the contract originator. Instead, these messages are always routed to a pre-determined ORDERERR destination.

  Note that for proper routing, the destination returned in the *C2 header must exist in the Output Handler Destination Table.
- SNS Entity ID – This optional field indicates the SNS entity that should process this transaction.
- Solicited/Unsolicited Indicator – For messages coming into SNS, this field has a value of "U." For messages going back to the contract originator, this field has a value of "S."
- Transaction Priority – This optional field indicates the priority level at which SNS should process this transaction.
- Message Format Indicator – This field always has a value of the numeric character zero (0).

3.4 Platform Header (*PLHDR) Section

The *PLHDR section is defined by the Provisioning PLATFORM and contains the information needed to perform security authorization checking and debugging. This section always immediately follows the control header in the FCIF message. Only those tags and options that are typically used in SNS are documented here. They include:

```
*PLHDR{
   USERID...;           User ID
   LCLID...;            Local ID <APP>
   DEBUG...;            Debugging flag [opt] <:FFF>
```

```
    TSO...;                  TSO data set [opt] <Y, TSO data
                             set name, DD:XXX>
    SUBSYS...;               Subsystem [opt] <OH>
    DUMP...;                 Dump flag [opt] <YES>
    DLI...;                  DLI flag [opt] <ALL>
}%
```

The USERID tag, which is required, identifies the end user (e.g., login ID) who initiated the request.

The LCLID tag (correlation code) is required for SNS to link to other non-MVS system applications and must always be set to "APP."

The remaining tags are optional and are used in conjunction with PLATFORM tracing facilities:

- The DEBUG tag turns on debugging and indicates the level of debugging output to be generated. A DEBUG value of ":FFF" is recommended to obtain the maximum amount of trace information.
- The TSO tag identifies the TSO data set to which the debugging output should be sent (when debugging is turned on).
  - When TSO is set to "Y," SNS allocates a file, USERID.DEBUG.
  - When TSO is set to a data set name, SNS allocates the specified data set name.
  - When TSO is set to DD:XXX (in which XXX is the DDNAME), SNS allocates the specified DDNAME.

  If the TSO tag is not specified, the debugging output defaults to the user terminal.
- The SUBSYS tag identifies the subsystem for which additional trace output is desired. Presently, "OH" (Output Handler) is the only subsystem option used by SNS.
- The DUMP tag causes a full system dump to occur when an error condition occurs in SNS.
- The DLI tag activates IMS DB/DC debugging. When DLI is set to "ALL", both IO and DB output are provided.

For additional information on debugging, refer to the "Troubleshooting" section of the *Service Negotiation Support (SNS) System Administration Guide.*

3.5 User Message (*UMSG) Section

The *UMSG section contains informational text that is related to fatal error conditions encountered by SNS or the PLATFORM's Immediate Message Poster (IMP), which controls processing flow for all FCIF messages entering SNS.

The data items for the *UMSG section are:

```
REC{                        Record
   MSGTEXT=...;             Message text
}
```

The MSGTEXT tag identifies the fatal error. The following messages may be returned:

Contract failed – Security not authorized

Contract failed – no entry in CONTRACT PGM MAP table

Contract failed – invalid mode in CONTRACT PGM MAP table

Contract failed – no entry in EXEC-ALIAS table

Contract failed – could not open memory file

Contract failed – could not run contract program

Contract failed – SNS encountered an internal program error and was unable to continue processing.

Contract failed – SNS encountered syntax errors in parsing the request message.

Contract failed – SNS could not find this session in its Session Database.

Contract failed – SNS could not find the destination for the message.

The *UMSG section is returned to the contract invoker only when SNS encounters an error while processing the original contract request. See the subsection titled "Exceptions to FCIF Message Structure Guidelines" in this section and "SNS Exception Handler" in the *Service Negotiation Support (SNS) System Administration Guide.*

3.5.1 SNS Header (*SNSHDR) Section

The *SNSHDR section contains information relevant to the entire message.

3.5.1.1 *SNSHDR Section: Request Message

The *SNSHDR may contain information needed by the invoking building block to continue processing the corresponding response message. ECHO and VDATA are optional tags supplied in the input message for this purpose. In most contracts, the *SNSHDR section is optional. However, the SNSNEG contract requires that the *SNSHDR be provided with the input date and time that the contract was invoked.

Data items for the *SNSHDR section in a *request* message are:

```
*SNSHDR{                     SNS Header
    ECHO=...;                Echo data [opt]
    VDATA=...;               Variable data [opt]
    IDATE{                   Input date [conditional - required in SNSNEG
                             contract]
        MONTH=...;           Input month
        DAY=...;             Input day
        YEAR=...;            Input year
    }                        End IDATE aggregate
    ITIME{                   Input time [conditional - required in SNSNEG
                             contract]
        HOUR=...;            Input hour
        MIN=...;             Input minutes
        SEC=...;             Input seconds
    }                        End ITIME aggregate
}%                           End *SNSHDR section
```

3.5.1.2 *SNSHDR Section: Response Message

On output, SNS always returns the *SNSHDR as it was originally received on input. These data items are identified in regular typeface. In addition, for negotiation dialogue contracts SNS may return the additional response data shown below in bold typeface.

```
*SNSHDR{                     SNS Header
    ECHO=...;                Echo data [opt]
    VDATA=...;               Variable data [opt]
    IDATE{                   Input date [conditional - required in SNSNEG
                             contract]
        MONTH=...;           Input month
```

```
          DAY=...;                  Input day
          YEAR=...;                 Input year
     }                              End IDATE aggregate
     ITIME{                         Input time [conditional - required in SNSNEG
                                    contract]
          HOUR=...;                 Input hour
          MIN=...;                  Input minutes
          SEC=...;                  Input seconds
     }                              End ITIME aggregate

     SNRC=...;                      Service negotiation return code {0-1}
     DATE{                          Date
          MONTH=...;                Month
          DAY=...;                  Day
          YEAR=...;                 Year
     }                              End DATE aggregate
     TIME{                          Time
          HOUR=...;                 Hour
          MIN=...;                  Minutes
          SEC=...;                  Seconds
     }                              End TIME aggregate
}%                                  End *SNSHDR section
```

SNS also returns a date/time stamp that indicates when the output message was formatted. This information is not returned when a fatal error is encountered.

For negotiation dialogue contracts, if SNS encounters a service request timing problem that prevents it from processing the entire contract, it also returns an SNS return code (SNRC) that identifies the problem. The possible values associated with the SNRC tag are:

SNRC	**Error condition**
001	A service request was not established in SNS.
002	A date/time stamp was not provided on input, but is required.
003	The input date/time stamp is invalid.

004 The service request was modified *after* this contract was invoked. Note that this condition is highly unlikely. However, when it occurs, SNS will not process the negotiation request(s) since different network resources may be involved.

005 A service request has already been established in SNS. The modify option should be used instead.

3.6 Exceptions to FCIF Message Structure Guidelines

If SNS encounters a fatal error condition while processing a request, it reports the problem to the error destination specified in the Output Handler Destination table.† This information is available to a system administrator, who can investigate the problem and take the appropriate course of action.

In certain situations, SNS notifies the contract invoker of the error condition. If SNS encounters an error while processing the original request from the contract invoker, it returns a response message back to the contract invoker with an indication of the problem. In this case, the response message contains the original request section (e.g., *C1, *RSNS, *ROSP, *RTNL) received on input, rather than the usual response section (e.g., *C2, *SNSR, *OSPR, *TNLR) that is returned under normal circumstances.

However, there are circumstances in which SNS is not able to provide this notification. If SNS encounters an error while processing a response from PREMIS or LFACS, it does not return a response to the contract invoker or perform its usual "clean-up" procedures (e.g., terminate the session). Furthermore, SNS is not able to notify the contract invoker of any updates that were made to LFACS.‡

† See the *Service Negotiation Support (SNS) System Administration Guide* for more information on the Output Handler Destination table.

‡ This is in conformance with the OSCA Contract Principle regarding "No Contract Accessibility Assumptions."

4. Negotiation Stand-Alone Contracts

The following stand-alone contracts assist in the negotiation of locally switched Digital Centrex and ISDN services:

- Street Address Validation Inquiry (INQSAG) – Determines whether a street address is valid in the SAG (Street Address Guide).
- Network Element Inquiry (INQNEI) – Identifies the network elements in the wire center associated with a street address.
- Product Marketing Inquiry (INQPMI) – Obtains product marketing information for a street address.
- Change in Service Inquiry (INQCHG) – Determines whether an existing circuit can support a regrade in service.
- Loop Availability Inquiry (INQLAI) – Determines whether a specified number of spare loops is available to support a particular product.
- Loop Reservation (UPDRSV) – Places or edits loop reservations.
- TN List Selection/Return (UPDTNL) – Selects or returns telephone numbers (TNs) associated with a regular Centrex group.

These contracts can be divided into three groups, based on their functions and FCIF message structure: Service Negotiation Support, Outside Plant, and TN List Selection and Return.

4.1 Service Negotiation Support Contracts

The Service Negotiation Support contracts assist the negotiator in inquiring about network availability. These contracts are INQSAG, INQNEI, INQPMI, and INQCHG. The structure of these contracts is outlined below.

A Service Negotiation Support contract contains a Service Negotiation Support Request FCIF message and a Service Negotiation Support Response FCIF message. The Service Negotiation Support Request FCIF message consists of the following sections†:

† The message sections are explained in detail later in this section.

*C1 section	Control Header section
*PLHDR section	Provisioning PLATFORM Header section
*SNSHDR section (optional)	SNS Header section
*RSNS section	Service Negotiation Support Request section

The Service Negotiation Support Response FCIF message consists of the following sections:

*C2 section	Control Header section
*PLHDR section	Provisioning PLATFORM Header section
*UMSG section (optional)	User Message section
*SNSHDR section (optional)	SNS Header section
*SNSR section	Service Negotiation Support Response section

4.2 Outside Plant Contracts

The Outside Plant contracts determine spare loop availability and perform loop reservation updates. These contracts are INQLAI and UPDRSV. The structure of these contracts is outlined below.

An Outside Plant Contract contains an Outside Plant Request FCIF message and an Outside Plant Response FCIF message. The Outside Plant Request FCIF message consists of the following sections:

*C1 section	Control Header section
*PLHDR section	Provisioning PLATFORM Header section
*SNSHDR section (optional)	SNS Header section
*ROSP section	Outside Plant Request section

The Outside Plant Response FCIF message consists of the following sections:

*C2 section	Control Header section
*PLHDR section	Provisioning PLATFORM Header section
*UMSG section (optional)	User Message section
*SNSHDR section (optional)	SNS Header section
*OSPR section	Outside Plant Response section

4.3 TN List Selection/Return Contract

The TN List Selection and Return contract, UPDTNL, supports the selection of Centrex TNs from an available pool and the return of selected TNs back to the pool. The structure of this contract is outlined below.

A TN List Selection/Return Contract contains a TN List Selection/Return Request FCIF message and a TN List Selection/Return Response FCIF message. The TN List Selection/Return Request FCIF message consists of the following sections:

*C1 section	Control Header section
*PLHDR section	Provisioning PLATFORM Header section
*SNSHDR section (optional)	SNS Header section
*RTNL section	TN List Selection/Return Request section

The TN List Selection/Return Response FCIF message consists of the following sections:

*C2 section	Control Header section
*PLHDR section	Provisioning PLATFORM Header section
*UMSG section (optional)	User Message section
*SNSHDR section (optional)	SNS Header section
*TNLR section	TN List Selection/Return Response section

Section 3 in this document describes the structure of the *C1, *C2, *PLHDR, *SNSHDR, and *UMSG sections. The remainder of this section explains the FCIF sections that are specific to the negotiation stand-alone contracts. (A description of each FCIF tag can be found in Appendix B.)

4.4 Request for Service Negotiation Support (*RSNS) Section

The *RSNS section is required in the request message of each of the following Service Negotiation Support contracts:

- Street Address Validation Inquiry (INQSAG)
- Network Element Inquiry (INQNEI)
- Product Marketing Inquiry (INQPMI)
- Change in Service Inquiry (INQCHG)

4.4.1 General *RSNS Section Format

A listing of all possible data items that could appear in the *RSNS section appears below. The data items listed below may be required, conditional or optional. Specific details about each data item are provided in the subsequent sections that describe each of the above contracts.

The data items for the *RSNS section are:

```
*RSNS{
  SVCLOC{                    Service location
     ACL{                    Action line
        SAGA=...;            SAG area
        ZIP=...;             Zip code
        ADDR{                Address
           BADR{             Basic address
                BAD=...;     Basic address designation
                AHN=...;     Assigned house number
                STR=...;     Street Name
                CNA=...;     Community name
                STN=...;     State name
           }
           SUPL{             Supplemental address
                SID=...;     Structure ID
                STYP=...;    Structure type
                EID=...;     Elevation ID
                ETYP=...;    Elevation type
                UID=...;     Unit ID
                UTYP=...;    Unit type
                }
            }
        }
    }

    CTL{                        Control data
        CTC=...;                Control code
        INQOPT=...;             Inquiry option
        SVCTYP=...;             Service type
    }

    ACL{                        Action line {0-1}
        MAINTN{                 Main TN
            NPA=...;            Main NPA
            NXX=...;            Main NXX
            LINE=...;           Main line
        }
        PROD=...;               Product name
        MADN=...;               MADN flag
        FO=...;                 Foreign office indicator
        WC=...;                 Wire center
        TN{                     Telephone number
            NPA=...;            NPA
            NXX=...;            NXX
            LINE=...;           Line
        }
        CLS=...;                CLCI™ serial number
        DPA=...;                Different premises address identifier
        CKL=...;                Circuit location identifier
        SIG=...;                Signaling OEC
    }

}%
```

COMMON LANGUAGE is a registered trademark and CLCI, CLEI, CLFI, and CLLI are trademarks of Bellcore.

The SVCLOC aggregate contains the living unit address that identifies the service location at which negotiation activity is to be performed. Appendix C provides additional information on how address data is passed in SNS contracts.

The CTL aggregate contains control information that indicates the type of processing that SNS should perform. It corresponds to the contract name in the FCIF control header. In particular, the control code (CTC) indicates the processing mode. The *inquiry* mode (CTC = I) is the only processing mode currently supported.

In the inquiry mode, the inquiry option (INQOPT) in the CTL aggregate identifies the type of inquiry to be performed:

- Street Address Validation Inquiry (INQOPT = SAG)
- Network Element Inquiry (INQOPT = NEI)
- Product Marketing Inquiry (INQOPT = PMI)
- Negotiation Inquiry (INQOPT = NEG)

The CTL aggregate may also contain the SVCTYP tag, which further identifies the type of service negotiation *inquiry* to be performed when INQOPT = NEG.

The ACL aggregate contains the specific information required to perform the requested inquiry. The information that SNS requires in each ACL aggregate varies, depending upon the INQOPT, SVCTYP, and CTC data that appear in the CTL aggregate.

The following subsections explain how the *RSNS section should be constructed, based on the type of Service Negotiation Support contract.

4.4.2 *RSNS Section Format: Street Address Validation Inquiry (INQSAG)

SNS uses the information provided in the *RSNS section of the INQSAG request message to determine whether the input street address falls within an address range in the PREMIS SAG.

As a reminder:

The *RSNS section is one section in the INQSAG request message. The structure of the sections in the INQSAG request is as follows:

*C1 section
*PLHDR section
*SNSHDR section
*RSNS section.

The following data items may appear in the *RSNS section:

```
*RSNS{
  SVCLOC{                  Service location
     ACL{                  Action line
        SAGA=...;          SAG area† [conditional – required if ZIP not
                           specified]
        ZIP=...;           Zip code† [conditional – required if SAGA not
                           specified]
        ADDR{              Address
           BADR{           Basic address
                BAD=...;   Basic address designation [opt]
                AHN=...;   Assigned house number [opt]
                STR=...;   Street name or descriptive address
                CNA=...;   Community name† [opt]
                STN=...;   State name† [opt]
           }
        }
     }
  }
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
  CTL{                       Control data
   CTC = I;                  Control code
   INQOPT = SAG;             Inquiry option
  }
}%
```

4.4.2.1 SNS Processing of INQSAG

Upon receiving an INQSAG request, SNS generates an APPVAL request to the PREMIS (Premises Information System) system to SAG-validate the input street address. If the street address is valid, PREMIS identifies the PREMIS wire center (WC) and supplies additional SAG and street-address-related information (e.g., primary street address). Otherwise, PREMIS may provide street address menu data to assist the negotiator in identifying the proper street address.

4.4.3 *RSNS Section Format: Network Element Inquiry (INQNEI)

SNS uses the information provided in the *RSNS section of the INQNEI request message to validate the input street address against the PREMIS SAG and to determine the network elements that exist in the associated wire center. If a MAINTN is specified, SNS identifies only the network element serving the customer, when this information is available in the SNS TNLIST database.

> As a reminder:
>
> The *RSNS section is one section in the INQNEI request message. The structure of the sections in the INQNEI request is as follows:
>
> *C1 section
> *PLHDR section
> *SNSHDR section
> *RSNS section.

The following data items may appear in the *RSNS section:

```
*RSNS{
  SVCLOC{                    Service location
     ACL{                    Action line
        SAGA=...;            SAG area† [conditional – required if ZIP not
                             specified]
        ZIP=...;             Zip code† [conditional – required if SAGA not
                             specified]
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
            ADDR{              Address
               BADR{           Basic address
                    BAD=...;   Basic address designation [opt]
                    AHN=...;   Assigned house number [opt]
                    STR=...;   Street name or descriptive address
                    CNA=...;   Community name† [opt]
                    STN=...;   State name† [opt]
               }
            }
         }
      }

      CTL{                     Control data
         CTC=I;                Control code
         INQOPT=NEI;           Inquiry option
      }

      ACL{                     Action line
         MAINTN{               Main TN [opt]
            NPA=...;           Main NPA
            NXX=...;           Main NXX
            LINE=...;          Main line
         }
      }
   }%
```

4.4.3.1 SNS Processing of INQNEI

Upon receiving an INQNEI request, SNS generates an APPSVA request to PREMIS to SAG-validate the input street address, as described in the INQSAG contract. In addition, if the address is SAG-valid, PREMIS identifies network elements located in the associated wire center. If a Centrex customer is specified in the input request, SNS determines the network element that serves that particular customer if this information is available in the SNS TNLIST database. SNS also supplies network element characteristics (e.g., switch type and generic), when available.

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

4.4.4 *RSNS Section Format: Product Marketing Inquiry (INQPMI)

SNS uses the information provided in the *RSNS section of the INQPMI request message to validate the input street address and to perform either of the following inquiries:

- Determine whether a particular product is marketed at the specified street address.
- Determine all products that are marketed at the specified street address (PROD=ALL).

In both cases, SNS provides a list of network elements that are capable of supporting each product. If a MAINTN is specified, SNS provides only product marketing information relevant to the network element serving the customer, when this information is available in the SNS TNLIST database. Foreign Office data is not returned unless the value of the Foreign Office indicator (FO) is Y † and the product is not found or is embargoed in the Local Serving Office.

> As a reminder:
>
> The *RSNS section is one section in the INQPMI request message. The structure of the sections in the INQPMI request is as follows:
>
> *C1 section
> *PLHDR section
> *SNSHDR section
> *RSNS section.

† In the case of an INQPMI request, FO assumes a value of Y unless specified otherwise.

The following data items may appear in the *RSNS section:

```
*RSNS{
  SVCLOC{                     Service location
     ACL{                     Action line
        SAGA=...;             SAG area† [conditional – required if ZIP not
                              specified]
        ZIP=...;              Zip code† [conditional – required if SAGA not
                              specified]
        ADDR{                 Address
           BADR{              Basic address
                BAD=...;      Basic address designation [opt]
                AHN=...;      Assigned house number [opt]
                STR=...;      Street name or descriptive address
                CNA=...;      Community name† [opt]
                STN=...;      State name† [opt]
           }
        }
     }
  }

  CTL{                        Control data
     CTC=I;                   Control code
     INQOPT=PMI;              Inquiry option
  }
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
  ACL{                   Action line
     PROD=...;           Product name <ALL, specific product>
     FO=...;             Foreign office indicator [opt] <Y,N>
     MAINTN{             Main TN [opt]
        NPA=...;         Main NPA
        NXX=...;         Main NXX
        LINE=...;        Main line
     }
  }
}%
```

4.4.4.1 SNS Processing of INQPMI

Upon receiving an INQPMI request, SNS generates a request to SAG-validate the input street address, as described in the INQSAG contract.

If a single product is specified on input, an attempt is made to determine if the specified product is being marketed at the input street address. If product=ALL, then all products available at that address are identified. In addition, the network element(s) (e.g. switch(es)) capable of supporting the product(s) is/are identified.

If the product is not found or is embargoed in the Local Serving Office, then the value of the Foreign Office Indicator is checked. If this value is Y (For the INQPMI request, the Foreign Office indicator assumes a default value of Y unless explicitly assigned a value of N on input), then the Foreign Office(s) are searched for the product(s) that are not found or are embargoed in Local Serving Office.

If a Main Account TN is not provided on input, all the information identified above is returned.

If a Main Account TN is specified on input, an attempt is made to map it to a network element in the SNS reference database. If the mapping is successful, an attempt is made to find the above network element among the network elements identified as being capable of supporting the product. If the search is successful, information about the network element and the product(s) it serves is returned.

4.4.5 *RSNS Section Format: Change in Service Inquiry (INQCHG)

SNS uses the information provided in the *RSNS section of the INQCHG request message to determine whether the specified product is available at the service address associated with an existing circuit. There are two parts to this inquiry: switch compatibility and loop compatibility. Switch compatibility is optional and can be bypassed by providing a wire center on input.

- Switch Compatibility: SNS validates the input street address. If product marketing information is available, SNS also determines whether the product is marketed at the specified address. If a main TN is provided and it can be mapped to a serving network element, SNS determines product availability for that network element. Foreign Office data is not returned unless the Foreign Office indicator (FO) is set to Y (unless explicitly set to Y on input, FO assumes a default value of N) and the product is not found or is embargoed in the Local Serving Office
- Loop Compatibility: SNS determines the loop transmission requirements, based on the values of the FO indicator, the MADN indicator,† if provided, and the product. An attempt is then made to determine whether the existing loop (identified by either a TN *or* CLS circuit identifier and an optional DPA *or* CKL circuit termination identifier) is associated with the specified service address and whether it can support the regrade in service based on the loop transmission requirements.

† If FO and MADN are both set to Y on input, MADN will be reset to N. If the user wishes for the MADN indicator to take on a value of Y, FO must be explicitly set to N on input.

> As a reminder:
>
> The *RSNS section is one section in the INQCHG request message. The structure of the sections in the INQCHG request is as follows:
>
> *C1 section
> *PLHDR section
> *SNSHDR section
> *RSNS section.

The following data items may appear in the *RSNS section:

```
*RSNS{

  SVCLOC{                 Service location
     ACL{                 Action line
        SAGA=...;         SAG area† [conditional – required if ZIP not
                          specified]
        ZIP=...;          Zip code† [conditional – required if SAGA not
                          specified]
        ADDR{             Address
           BADR{          Basic address
                BAD=...;  Basic address designation [opt]
                AHN=...;  Assigned house number [opt]
                STR=...;  Street name or descriptive address
                CNA=...;  Community name† [opt]
                STN=...;  State name† [opt]
           }

           SUPL{           Supplemental address
                SID=...;   Structure ID [opt]
                STYP=...;  Structure type [opt]
                EID=...;   Elevation ID [opt]
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
                  ETYP=...;     Elevation type [opt]
                  UID=...;      Unit ID [opt]
                  UTYP=...;     Unit type [opt]
              }
          }
      }
  }

  CTL{                          Control data
      CTC=I;                    Control code
      INQOPT=NEG;               Inquiry option
      SVCTYP=C;                 Service type
  }

  ACL{                          Action line
      PROD=...;                 Product name
      MAINTN{                   Main TN [opt]
          NPA=...;              Main NPA
          NXX=...;              Main NXX
          LINE=...;             Main line
      }
      MADN=...;                 MADN flag [opt] <Y>
      FO=...;                   Foreign office indicator [opt] <Y,N>
      WC=...;                   Wire center [opt]
      TN{                       Telephone number [conditional – required if CLS
                                not specified]
          NPA=...;              NPA
          NXX=...;              NXX
          LINE=...;             Line
      }
      CLS=...;                  CLCI serial number [conditional – required if TN
                                not specified]
      DPA=...;                  Different premises address identifier [opt]
      CKL=...;                  Circuit location identifier [opt]
      SIG=...;                  Signaling OEC [opt – default = L] <L, G or B>
  }
}%
```

4.4.5.1 SNS Processing of INQCHG

There are two parts to this inquiry: switch compatibility and loop compatibility. Switch compatibility is optional and can be bypassed by providing a wire center on input.

Switch Compatibility: On receiving an INQCHG request, SNS generates a request to SAG-validate the input street address and determine product availability, as described in the INQPMI contract.

Foreign Office data is not returned unless the value of the Foreign Office indicator (FO) is **Y** (for the INQCHG request, FO assumes a default value of **N** unless explicitly set to **Y** on input) and the product is not found or is embargoed in the Local Serving Office. If a main TN is provided on input, an attempt is made to map it to a network element and product availability is determined for that network element. If the specified product is not available, the loop compatibility portion of the processing is bypassed and the above results are returned to the requesting system in a response.

Loop Compatibility: SNS determines the loop transmission requirements (based on the values of the Foreign Office indicator, the MADN indicator,† if provided, and the product) needed to support the service regrade (Centrex). Next, the living unit address and circuit identifier are validated and any discrepancies found are reported to SNS. Otherwise, the working circuit currently assigned to the customer is examined to determine if the loop facilities meet the transmission requirements.

† If FO and MADN are both set to Y on input, MADN will be reset to N. If the user wishes for the MADN indicator to take on a value of Y, FO must explicitly be set to N on input.

4.5 Service Negotiation Support Response (*SNSR) Section

The *SNSR section returned in the response message indicates the results of the Service Negotiation Support inquiry contract. It may contain address and network-related information, as well as identify errors and discrepancies.

4.5.1 General *SNSR Section Format

The data items† for the *SNSR section are shown in the following general format. The items shown in regular typeface reflect the data provided in the original request (*RSNS section). The original input data from the *RSNS section of the request message is always provided in the corresponding *SNSR section of the response message. (The original input data shown in this general format does not indicate multiple occurrences of data items, since that depends on the information provided in the request message.) The items shown in bold typeface represent the response data that SNS supplies. The first group of response data – from IERC through LPRC – includes the CTL aggregate return codes. The second group of response data – from SAGAM through CKTM – shows the additional data aggregates that may appear in the *SNSR section.‡ The return codes and the additional data aggregates are explained in the sections that follow the general format.

As a reminder:

The *SNSR section is one section in the response message. The structure of the sections in the response message is as follows:

*C2 section
*PLHDR section
*SNSHDR section
*SNSR section.

† Data items shown in the *general* section format do not indicate whether they are optional, conditional or required. This information is supplied in the subsequent sections.

‡ In an actual *SNSR section, each of these additional data aggregates would have data items under it. These data items are not shown in the general format; however, they are explained in subsequent sections.

```
*SNSR{
  SVCLOC{                        Service location
     ACL{                        Action line
        SAGA=...;                SAG area
        ZIP=...;                 Zip code
        ADDR{                    Address
           BADR{                 Basic address
                BAD=...;         Basic address designation
                AHN=...;         Assigned house number
                STR=...;         Street name
                CNA=...;         Community name
                STN=...;         State name
           }
           SUPL{                 Supplemental address
                SID=...;         Structure ID
                STYP=...;        Structure type
                EID=...;         Elevation ID
                ETYP=...;        Elevation type
                UID=...;         Unit ID
                UTYP=...;        Unit type
           }
        }
     }
  }

  CTL{                          Control data
     CTC=...;                   Control code
     INQOPT=...;                Inquiry option
     SVCTYP=...;                Service type

     IERC=...;                  Input error return code {0-N}, where N = the
                                number of input errors detected
     SARC=...;                  Street address return code {0-1}
     LURC=...;                  Living unit return code {0-1}
     NERC=...;                  Network element return code {0-P}, where P =
                                the number of network element conditions to
                                report
```

```
    PMRC=...;               Product marketing return code {0-M}, where M
                            = the number of product marketing conditions to
                            report
    LPRC=...;               Loop return code {0-1}
}

ACL{                        Action line {0-1}
    MAINTN{                 Main TN
        NPA=...;            Main NPA
        NXX=...;            Main NXX
        LINE=...;           Main line
    }
    PROD=...;               Product name
    MADN=...;               MADN flag
    FO=...;                 Foreign office indicator
    WC=...;                 Wire center
    TN{                     Telephone number
        NPA=...;            NPA
        NXX=...;            NXX
        LINE=...;           Line
    }
    CLS=...;                CLCI serial number
    DPA=...;                Different premises address identifier
    CKL=...;                Circuit location identifier
    SIG=...;                Signaling OEC
}

SAGAM{                      SAGA menu {0-1}
}
STRM{                       Street name menu {0-1}
}
DESCRIPM{                   Descriptive name menu {0-1}
}
ARGM{                       Address range menu {0-1}
}
ADDRM{                      Address menu {0-1}
}
STRARGM{                    Street name/address range menu {0-1}
}
```

```
   BADRM{                 Basic address menu {0-1}
   }
   PARDESCRIP{            Descriptive partial match {0-1}
   }
   STRINFO{               Street address information {0-1}
   }
   SAGINFO{               SAG information {0-1}
   }
   LUINFO{                LU information {0-1}
   }
   SIMSTRPT{              Similar street report {0-1}
   }
   LUM{                   Living unit menu {0-1}
   }
   NEINFO{                Network element information {0-1}
   }
   PMINFO{                Product market information {0-1}
   }
   CKTM{                  Circuit menu {0-1}
   }
}%
```

4.5.2 CTL Aggregate Return Codes

There are six return codes that may be included with the CTL aggregate data. In the general format above, they are shown in bold typeface under the CTL aggregate. These return codes are:

- *Input Error Return Code*

  The contract invoker is responsible for providing SNS with a valid Service Negotiation Support request message that conforms to the specifications outlined in this document. However, if SNS detects a critical *input error* condition that prevents it from continuing its processing, it notifies the contract invoker of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times for multiple input errors.

- *Street Address Return Code*

  The SARC tag indicates the validity of the street address provided on input. It is produced whenever the Service Negotiation Support contract involves a request to PREMIS to SAG-validate the input street address.

- *Living Unit Return Code*

  The LURC tag indicates the validity of the living unit address sent to LFACS. It is produced whenever an INQCHG contract involves a request to LFACS (*except* when loop transmission requirements could not be determined).

- *Network Element Return Code*

  The NERC tag indicates whether network elements were identified for a SAG-valid input street address. It is produced on an INQNEI contract only and may appear multiple times.

- *Product Marketing Return Code*

  The PMRC tag indicates product marketing conditions at a SAG-valid input street address. It is produced on an INQPMI or INQCHG contract only and may appear multiple times.

- *Loop Return Code*

  The LPRC tag indicates the results of *loop processing* in LFACS. It is produced on an INQCHG contract only.

The specific return codes that may appear in the *SNSR section depend on the type of contract. Table 4-1 shows which return codes are possible for each of the four Service Negotiation Support contracts. (Refer to Appendix D for a description of each return code value.)

Table 4-1. Service Negotiation Support Contract Return Codes

Contract	IERC	SARC	LURC	LPRC	PMRC	NERC
INQSAG	X	X				
INQNEI	X	X				X
INQPMI	X	X			X	
INQCHG	X	X	X	X	X	

4.5.3 Additional Data Aggregates Provided in *SNSR Section

There are 16 additional data aggregates that may appear in the *SNSR section. In the general format shown previously, they appear in bold typeface at the end of the *SNSR section. These additional data aggregates are included only when specific return codes and values appear in the CTL aggregate. Table 4-2 shows the relationships between return code values and additional data aggregates. The individual data items that would appear with the additional data aggregates are described in the sections that follow the table.

Table 4-2. *SNSR CTL Return Codes and Related Data Aggregates

Return Code	Value	Related Data Aggregate
SARC	010	SAGAM
SARC	004	STRM
SARC	006	DESCRIPM
SARC	007,014	ARGM
SARC	008	ADDRM
SARC	009	STRARGM
SARC	012,013	BADRM
SARC	013	PARDESCRIP
SARC	> 100	SAGINFO
SARC	102-106	STRINFO
LURC	004	SIMSTRPT
LURC	003	LUM
LURC	002	LUINFO
NERC	002,004	NEINFO
PMRC	001,001-1, 003,003-1, 004,004-1 006,006-1, 010,010-1	PMINFO
LPRC	002	CKTM

4.5.3.1 *SNSR Section Format: SAG Area Menu

The following identifies the *SNSR data items returned when the input zip code is associated with multiple SAG Areas and SAGA was not provided on input. A SAGAM aggregate is returned when the following return code appears in the CTL aggregate:

SARC = 010.

```
SAGAM{                      SAGA menu
       SAGA=...;            SAG area {1 - N}
}
```

In the SAGA line, "N" is the number of SAGAs found in PREMIS that are associated with the input zip code.

4.5.3.2 *SNSR Section Format: Street Name Menu for More than One Partial Match

The following identifies the *SNSR data items returned when more than one partial match is found on the input street name. An STRM aggregate is returned when the following return code appears in the CTL aggregate:

SARC = 004.

```
STRM{                       Street name menu
       SIMSTR{              Similar street name {2-N}, where N=the number
                            of streets that partially match the input street name
          STR=...;          Street name
          ARG{              Address range
                LHV=...;    Low house number value
                HHV=...;    High house number value
                RGIN=...;   Range indicator
          }
          CNA=...;          Community name
          STN=...;          State name
          ZIP=...;          Zip code [opt]
       }
}
```

4.5.3.3 *SNSR Section Format: Descriptive Name Menu for More than One Partial Match

The following identifies the *SNSR data items returned when more than one partial match is found on the input descriptive. A DESCRIPM aggregate is returned when the following return code appears in the CTL aggregate:

SARC = 006.

```
DESCRIPM{                       Descriptive name menu
    SIMDESCRIP{                 Similar descriptive name {2-N}, where N=the
                                number of SNS descriptive addresses that partially
                                match the descriptive address input
        DESCRIP=...;            Descriptive address
        BAD=...;                Basic address designation
        AHN=...;                Assigned house number
        STR=...;                Street name
        CNA=...;                Community name
        SRMK1=...;              SAG remark 1 [opt]
        SRMK2=...;              SAG remark 2 [opt]
        SRMK3=...;              SAG remark 3 [opt]
    }
}
```

4.5.3.4 *SNSR Section Format: House Number/AHN Number Menu

The following identifies the *SNSR data items returned when particular address verification errors have been determined. An ARGM aggregate is returned when one of the following return codes appears in the CTL aggregate:

SARC = 007 or 014.

```
ARGM{                       Address range menu
    ARGDA{                  Address range data {1 - N}, where N= the number
                            of all valid address ranges for the input street name
      ARG{                  Address range
          LHV=...;          Low house number value
          HHV=...;          High house number value
          RGIN=...;         Range indicator
      }
      CNA=...;              Community name
      STN=...;              State name
      ZIP=...;              Zip code [opt]
    }
}
```

4.5.3.5 *SNSR Section Format: Address Duplicated (Community/State Menu)

The following identifies the *SNSR data items returned when the address is duplicated in more than one community/state. An ADDRM aggregate is returned when the following return code appears in the CTL aggregate:

SARC = 008.

```
ADDRM{                        Address menu
    ADDRDA{                   Address data {2 - N} where N= the number of
                              duplicate SAG records for the input address
      CNA=...;                Community name
      CNA1=...;               Alternate community name 1 [opt]
      CNA2=...;               Alternate community name 2 [opt]
      STN=...;                State name
      ZIP=...;                Zip code
      ARG{                    Address range
          LHV=...;            Low house number value
          HHV=...;            High house number value
          RGIN=...;           Range indicator
      }
    }
}
```

A community name value may occur as a primary community for one address area and as an alternate community for another address area. Since these address areas may contain duplicate basic addresses, SNS needs to be alerted when a Request for Service Negotiation Support (*RSNS) contains a community name derived from this menu. Therefore, PREMIS has inserted the string $$$ at the beginning of each CNA value on this menu. The contract invoker must return this string ($$$) to SNS in the CNA value on any subsequent request to provide community uniqueness and to avoid receiving the ADDRM menu again.

4.5.3.6 *SNSR Section Format: Street Name/Address Range Menu

The following identifies the *SNSR data items returned when a particular address verification error has been determined. A STRARGM aggregate is returned when the following return code appears in the CTL aggregate:

SARC = 009.

```
STRARGM{                      Street name/address range menu
    SIMSTR{                   Similar street name {1 - N}, where N = the number
                              of street names that partially match the input street
                              name
      STR = ...;              Street name
      ARG{                    Address range
          LHV = ...;          Low house number value
          HHV = ...;          High house number value
          RGIN = ...;         Range indicator
      }
      CNA = ...;              Community name
      STN = ...;              State name
      ZIP = ...;              Zip code [opt]
    }
}
```

4.5.3.7 *SNSR Section Format: Select Basic Address Menu

The following identifies the *SNSR data items returned when more than one basic address is associated with the input descriptive. A BADRM aggregate is returned when one of the following return codes appears in the CTL aggregate:

SARC = 012 or 013.

```
BADRM{                   Basic address menu
    BADRDA{              Basic address data {2 - N}, where N=  the number
                         of basic addresses that are associated with the input
                         descriptive address
      BAD=...;           Basic address designation† [opt]
      AHN=...;           Assigned house number† [opt]
      STR=...;           Street name
      CNA=...;           Community name
      SRMK1=...;         SAG remark 1 [opt]
      SRMK2=...;         SAG remark 2 [opt]
      SRMK3=...;         SAG remark 3 [opt]
    }
}
```

† Only a BAD or an AHN will be output as part of the basic address data.

4.5.3.8 *SNSR Section Format: Partial Match on Descriptive

The following identifies the *SNSR data items returned when an exact match is not found on the input descriptive and only one partial match is found, but more than one basic address is associated with it.

A PARDESCRIP aggregate is returned when the following return code appears in the CTL aggregate:

SARC = 013.

```
PARDESCRIP{                 Descriptive partial match
    PSAGA=...;              PREMIS SAG area
    PADR{                   PREMIS address area
        BADR{               Basic address
            STR=...;        Street name
        }
    }
}
```

4.5.3.9 *SNSR Section Format: Similar Street Report

The following identifies the *SNSR data items returned when a street address verification error occurs in LFACS and more than one partial match is found. A SIMSTRPT aggregate is returned when the following return code appears in the CTL aggregate:

LURC = 004.

```
SIMSTRPT{                     Similar street report
     TEXT=...;                Similar street report text
}
```

4.5.3.10 *SNSR Section Format: Living Unit Menu

The following identifies the *SNSR data items returned when a living unit verification error occurs in LFACS and a neighborhood report of nearby living units is provided. A LUM aggregate is returned when the following return code appears in the CTL aggregate:

LURC = 003.

```
LUM{                          Living unit menu
     TEXT=...;                Living unit report text
}
```

4.5.3.11 *SNSR Section Format: SAG Information

The following identifies the *SNSR SAG-related data items returned for a SAG-valid input street address. A SAGINFO aggregate is returned when any of the following return codes appears in the CTL aggregate:

SARC = all return codes greater than 100.

```
SAGINFO{                   SAG information
     SAGA=...;             SAG area
     CNA1=...;             Alternate community name 1 [opt]
     CNA2=...;             Alternate community name 2 [opt]
     ALT=...;              Alternate address [opt]
     DESCRIP=...;          Descriptive address [opt]
     EXCH=...;             Exchange
     WC=...;               Wire center
     RTZ=...;              Rate zone [opt]
     TAR=...;              Tax area [opt]
     ZIP=...;              Zip code [opt]
     NPA=...;              NPA
     TELF=...;             Telephone features [opt]
     BO=...;               Business office [opt]
     DIR=...;              Directory [opt]
     PD=...;               Plant district [opt]
     PC=...;               Phone center [opt]
     CO=...;               Central office [opt]
     LCL=...;              Local geographic area information [opt]
     SRMK1=...;            SAG remark 1 [opt]
     SRMK2=...;            SAG remark 2 [opt]
     SRMK3=...;            SAG remark 3 [opt]
     CNA=...;              Community name
     CNARQ=...;            Community name required indicator [opt]
     STN=...;              State name
     STNRQ=...;            State name required indicator [opt]
}
```

4.5.3.12 *SNSR Section Format: Modified Street Information

The following identifies the *SNSR data items returned when the input street address was "modified" to find a match in PREMIS. It identifies the *primary* street address found in PREMIS. A STRINFO aggregate is returned when any of the following return codes appears in the CTL aggregate:

SARC = 102 through 106.

```
STRINFO{                    Street address information
     STRADR{                Street address
          BADR{             Basic address
               BAD=...;     Basic address designation† [opt]
               AHN=...;     Assigned house number† [opt]
               STR=...;     Street name
          }
     }
}
```

† A BAD or an AHN will always be output as part of the basic address data.

4.5.3.13 *SNSR Section Format: Modified Living Unit Information

The following identifies the *SNSR data items returned when the PREMIS (or input) street address and/or input SUPL were modified to find a living unit match in LFACS. A LUINFO aggregate is returned when the following return code appears in the CTL aggregate:

LURC = 002.

```
LUINFO{                         Living unit information
     LUADR{                     Living unit address
         BADR{                  Basic address
              BAD=...;          Basic address designation
              AHN=...;          Assigned house number
              STR=...;          Street name
              CNA=...;          Community name
              STN=...;          State name
         }
         SUPL{                  Supplemental address
              SID=...;          Structure ID
              STYP=...;         Structure type
              EID=...;          Elevation ID
              ETYP=...;         Elevation type
              UID=...;          Unit ID
              UTYP=...;         Unit type
         }
     }
}
```

4.5.3.14 *SNSR Section Format: Network Element Information

The following identifies the *SNSR data items returned once SNS has determined the network elements that exist in the wire center associated with the input street address. If a MAINTN was specified on input, SNS identifies only the network element serving the customer, when this information is available in the TNLIST database.

A NEINFO aggregate is returned when one of the following return codes appears in the CTL aggregate:

NERC = 002 or 004.

```
NEINFO{                       Network element information
     SOD{                     Serving office data
          NE{                 Network element {1 - N}, where N = the number
                              of network elements that exist in the wire center
                              associated with the street address OR where N = 1
                              to identify the network element serving a particular
                              customer
               CLLI=...;      Network element CLLI code
               NETYP=...;     Network element type [opt]
               NEGEN=...;     Network element generic [opt]
          }
     }
}
```

4.5.3.15 *SNSR Section Format: Product Marketing Information

The following identifies the *SNSR data items returned once SNS has: (1) determined that a specific product is marketed at a street address, or (2) found all products that are marketed at a street address. It also contains information about the network elements capable of providing each product. If a MAINTN was specified on input, SNS filters the information pertinent to that customer only (when possible).

A PMINFO aggregate is returned when any of the following return codes appears in the CTL aggregate:

PMRC = 001, 001-1, 003, 003-1, 004, 004-1, 006, 006-1, 010 and 010-1.

Serving office data (SOD) is supplied only when PMRC = 001, 001-1, 003, 003-1, 006, 006-1.

Foreign office data (FOD) is supplied only under the following conditions: (1) a particular product and FO were specified on input and the PMRC = 004, 004-1, or (2) PROD=ALL and FO were specified on input and the PMRC = 010, 010-1.

```
PMINFO{                       Product market information
    SOD{                      Serving office data {0-1}
        MKTPROD{              Marketed product {1 - K}, where K = the number
                              of products marketed at the address OR where K =
                              1 when a specific product is specified on input
            PROD=...;         Product name
            EMB{              Embargo data {1 - 2} [opt]
              EMBLVL=...;     Embargo level <GSG,RNG>
              RELDATE{        Relief date {0-1}
                YEAR=...;     Year
                MONTH=...;    Month
                DAY=...;      Day
              }
              EMBRMK=...;     Embargo reason
            }
            NE{               Network element {1 - M}, where M = the number
                              of network elements in the serving office that can
                              provide the service
              CLLI=...;       Network element CLLI code
              NETYP=...;      Network element type [opt]
              NEGEN=...;          Network element generic [opt]
              EMBFLG=...;         Embargo Flag <Y>
              RELDATE{            Relief date {0-1}
                YEAR=...;         Year
                MONTH=...;        Month
                DAY=...;          Day
              }
              EMBRMK=...;         Embargo reason
            }
        }
    }
    FOD{                          Foreign Office Data {1 - N} where N = the
                                  number of foreign offices that can provide the
                                  specified product
        WC=...;                   Wire center
        TTA=...;                  Terminating traffic area
        MKTPROD{                  Marketed product {1 - P} where P = the number
                                  of available products supported by the foreign
                                  office
            PROD=...;             Product Name
            NE{                   Network element {1 - R}, where R = the number
                                  of network elements in the serving office that can
                                  provide the service
              CLLI=...;           Network element CLLI code
              NETYP=...;          Network element type [opt]
              NEGEN=...;          Network element generic [opt]
              EMBFLG=...;         Embargo Flag <Y>
              RELDATE{            Relief date {0-1}
                YEAR=...;         Year
                MONTH=...;        Month
                DAY=...;          Day
              }
              EMBRMK=...;         Embargo reason
            }
        }
    }
}
```

4.5.3.16 *SNSR Section Format: Circuit Menu

The following identifies the *SNSR data items returned when a circuit ID discrepancy occurs. A CKTM aggregate is returned when the following return code appears in the CTL aggregate:

LPRC = 002.

```
CKTM{                        Circuit menu
     TEXT=...;               Circuit report text
}
```

4.6 Request for Outside Plant (*ROSP) Section

The *ROSP section is required in the request message of each Outside Plant contract:

- Loop Availability Inquiry (INQLAI)
- Loop Reservation Update (UPDRSV).

4.6.1 General *ROSP Section Format

The data items[†] for the *ROSP section are:

```
*ROSP{
  SVCLOC{                        Service location
     ACL{                        Action line
         SAGA=...;               SAG area
         ZIP=...;                Zip code
         ADDR{                   Address
              BADR{              Basic address
                BAD=...;         Basic address designation
                AHN=...;         Assigned house number
                STR=...;         Street name or descriptive address
                CNA=...;         Community name
                STN=...;         State name
              }
              SUPL{              Supplemental address
                SID=...;         Structure ID
                STYP=...;        Structure type
                EID=...;         Elevation ID
                ETYP=...;        Elevation type
                UID=...;         Unit ID
                UTYP=...;        Unit type
              }
         }
```

[†] Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
      }
  }

  CTL{                        Control data
      CTC=...;                Control code
  }

  ACL{                        Action line {1-2}
      ACT=...;                Action code
      PROD=...;               Product name
      MADN=...;               MADN flag
      FO=...;                 Foreign office indicator
      WC=...;                 Wire center
      NUMLP=...;              Number of loops
      ACCT=...;               Customer account
      MKSEG=...;              Market segment indicator
      TN{                     Telephone number
          NPA=...;            NPA
          NXX=...;            NXX
          LINE=...;           Line
      }
      CLS=...;                CLCI serial number
      DPA=...;                Different premises address identifier
      CKL=...;                Circuit location identifier
      PFO=...;                Planned facilities order
      SO=...;                 Service order
      SIG=...;                Signaling OEC
      RSVDATE{                Reservation due date
          YEAR=...;           Year
          MONTH=...;          Month
          DAY=...;            Day
      }
      RSVRMK=...;             Reservation remark
      NOTICE=...;             Conference notice indicator
  }

}%
```

The SVCLOC aggregate contains the living unit address that identifies the service location at which negotiation activity is to be performed. Appendix C provides additional information on how address data is passed in SNS contracts.

The CTL aggregate contains control information. In particular, the control code (CTC) is used to determine the type of loop activity to be performed:

- Inquire about spare loop availability (CTC = I)
- Establish/build a loop reservation (CTC = B)
- Change a loop reservation (CTC = C)
- Remove a loop reservation (CTC = R)

The ACL aggregate contains the specific information required to perform the loop inquiry or reservation update. The information that SNS requires in each ACL aggregate varies, depending upon the CTC that appears in the CTL aggregate.

On a loop reservation update, the action code (ACT) within the ACL aggregate identifies the type of ACL: old or new. The following types of ACLs are required, based on the CTC value:

- When CTC = B, only a new ACL is required.
- When CTC = C, both an old and a new ACL are required.
- When CTC = R, only an old ACL is required.

The following subsections explain how the *ROSP section should be constructed, depending on the type of control data provided in the CTL aggregate.

4.6.2 *ROSP Section Format: Loop Availability Inquiry (INQLAI)

SNS uses the information provided in the *ROSP section of the INQLAI request message to determine whether a specified number of loops are available to support a particular product at a service location. If the WC is not specified in the input request, SNS determines it by validating the street address in PREMIS. Once the WC is known, SNS determines the loop transmission requirements based on the product, MADN indication, and whether or not the service is locally switched (specified by the foreign office indicator FO on input).† SNS then uses this information to inquire about spare loops in LFACS. ACCT and MKSEG are other optional criteria that LFACS may consider when identifying spare loops.

As a reminder:

The *ROSP section is one section in the INQLAI request message. The structure of the sections in the INQLAI request is as follows:

- *C1 section
- *PLHDR section
- *SNSHDR section
- *ROSP section.

† If FO and MADN are both set to Y on input, MADN will be reset to N. If the user wishes for the MADN indicator to take on a value of Y, FO must be explicitly set to N on input.

The following data items may appear in the *ROSP section:

```
*ROSP{

  SVCLOC{                     Service location
     ACL{                     Action line
        SAGA=...;             SAG area† [conditional – required if ZIP not
                              specified]
        ZIP=...;              Zip code† [conditional – required if SAGA not
                              specified]
        ADDR{                 Address
           BADR{              Basic address
                BAD=...;      Basic address designation [opt]
                AHN=...;      Assigned house number [opt]
                STR=...;      Street name or descriptive address
                CNA=...;      Community name† [opt]
                STN=...;      State name† [opt]
           }
           SUPL{              Supplemental address
                SID=...;      Structure ID [opt]
                STYP=...;     Structure type [opt]
                EID=...;      Elevation ID [opt]
                ETYP=...;     Elevation type [opt]
                UID=...;      Unit ID [opt]
                UTYP=...;     Unit type [opt]
           }
        }
     }
  }

  CTL{                        Control data
     CTC=I;                   Control code
  }
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
  ACL{                     Action line
      PROD=...;            Product name
      MADN=...;            MADN flag [opt] <Y>
      FO=...;              Foreign office indicator [opt] <Y>
      WC=...;              Wire center [opt]
      NUMLP=...;           Number of loops
      SIG=...;             Signaling OEC [opt - default = L] <L, G or B>
      ACCT=...;            Customer account [opt]
      MKSEG=...;           Market segment indicator [opt] <R, P, M, C, G, I, or A>
  }
}%
```

4.6.2.1 SNS Processing of INQLAI

Upon receiving an INQLAI request, SNS generates an APPVAL request to PREMIS to SAG-validate the input address, as described in the INQSAG contract (unless the WC is provided in the input request). SNS determines the loop transmission requirements (based on the values of the product, the Foreign Office indicator and the MADN indicator,† if provided) needed to support the requested product. Using this information, SNS generates an INQRES request to LFACS to determine whether the requested number of spare loops exists. As part of its processing, LFACS also validates the living unit address and identifies address discrepancies.

† If FO and MADN are both set to Y on input, MADN will be reset to N. If the user wishes for the MADN indicator to take on a value of Y, FO must be explicitly set to N on input.

4.6.3 *ROSP Section Format: Build Loop Reservation (UPDRSV)

SNS uses the information provided in the *ROSP section of the UPDRSV request message to establish a reservation for a specified number of loops capable of supporting a particular product at a service location. If the WC is not specified in the input request, SNS determines it by validating the street address in PREMIS. Once the WC is known, SNS determines the loop transmission requirements based on the product, MADN, and whether or not the service is locally switched (specified by the foreign office indicator FO on input).† SNS then establishes a loop reservation in LFACS.

A loop reservation may be established at a living unit, using any *one* of the following reservation keys:

- A single loop can be reserved for an individual circuit, identified by either a TN *or* a CLS circuit identifier. A DPA *or* CKL circuit termination identifier is optional.
- Any number of loops can be reserved for a planned facilities order (PFO).
- Any number of loops can be reserved for a service order (SO).

When multiple reservations are established with the same reservation key, it is recommended that they apply to loops with the same transmission requirements. Should any subsequent modifications be made to the reservation, LFACS replaces or removes the most recently reserved loops first. It does NOT take the loop transmission requirements into account when identifying these loops.

The ACCT and MKSEG data items are other optional criteria that LFACS may consider when reserving loops. If the NOTICE indicator is set, LFACS sends a confirmation notice to impacted organizations.

† If FO and MADN are both set to Y on input, MADN will be reset to N. If the user wishes for the MADN indicator to take on a value of Y, FO must be explicitly set to N on input.

> As a reminder:
>
> The *ROSP section is one section in the UPDRSV request message. The structure of the sections in the UPDRSV request is as follows:
>
> *C1 section
> *PLHDR section
> *SNSHDR section
> *ROSP section.

The following data items may appear in the *ROSP section:

```
*ROSP{

 SVCLOC{                 Service location
    ACL{                 Action line
       SAGA=...;         SAG area† [conditional – required if ZIP not
                         specified]
       ZIP=...;          Zip code† [conditional – required if SAGA not
                         specified]
       ADDR{             Address
          BADR{          Basic address
               BAD=...;  Basic address designation [opt]
               AHN=...;  Assigned house number [opt]
               STR=...;  Street name or descriptive address
               CNA=...;  Community name† [opt]
               STN=...;  State name† [opt]
          }

          SUPL{           Supplemental address
               SID=...;   Structure ID [opt]
               STYP=...;  Structure type [opt]
               EID=...;   Elevation ID [opt]
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
                ETYP = ...;      Elevation type [opt]
                UID = ...;       Unit ID [opt]
                UTYP = ...;      Unit type [opt]
            }
        }
    }
}

CTL{                             Control data
    CTC = B;                     Control code
}

ACL{                             New action line
    ACT = N;                     Action code
    PROD = ...;                  Product name
    MADN = ...;                  MADN flag [opt] < Y >
    FO = ...;                    Foreign office indicator [opt] < Y >
    WC = ...;                    Wire center [opt]
    NUMLP = ...;                 Number of loops [conditional − required when PFO or
                                 SO specified]
    ACCT = ...;                  Customer account [opt]
    MKSEG = ...;                 Market segment indicator [opt] < R, P, M, C, G, I, or A >
    TN{                          Telephone number [conditional − required if CLS, PFO
                                 or SO not specified]
        NPA = ...;               NPA
        NXX = ...;               NXX
        LINE = ...;              Line
    }
    CLS = ...;                   CLCI serial number [conditional − required if TN, PFO,
                                 or SO not specified]
    DPA = ...;                   Different premises address identifier [opt]
    CKL = ...;                   Circuit location identifier [opt]
    PFO = ...;                   Planned facilities order [conditional − required if TN,
                                 CLS, or SO not specified]
    SO = ...;                    Service order [conditional − required if TN, CLS, or PFO
                                 not specified]
    SIG = ...;                   Signaling OEC [opt − default = L] < L, G or B >
    RSVDATE{                     Reservation due date
        YEAR = ...;              Year
```

```
            MONTH=...;           Month
            DAY=...;             Day
        }
        RSVRMK=...;              Reservation remark [opt]
        NOTICE=...;              Confirmation notice indicator [opt]
    }
}%
```

4.6.3.1 SNS Processing of UPDRSV for Build

Upon receiving a UPDRSV request, SNS generates an APPVAL request to PREMIS to SAG-validate the input address, as described in the INQSAG contract (unless the WC is provided in the input request). If the PREMIS SAG validation is successful, SNS uses the PREMIS primary street address to construct the living unit address. Otherwise, SNS uses the input street address. In either case, SNS uses the input supplemental address data (LOC) to construct the living unit address. SNS then generates the following UPDRES request to LFACS for this living unit:

- Establish a loop reservation

  SNS determines the loop transmission requirements (based on the values of the product, the Foreign Office indicator and the MADN indicator,† if provided) needed to support the requested ISDN or Centrex product and then generates a request to LFACS to reserve one or more loops.

† If FO and MADN are both set to Y on input, MADN will be reset to N. If the user wishes for the MADN indicator to take on a value of Y, FO must be explicitly set to N on input.

4.6.4 *ROSP Section Format: Remove Loop Reservation (UPDRSV)

SNS uses the information provided in the *ROSP section of the UPDRSV request message to remove a loop reservation established earlier in LFACS. If the WC is not specified in the input request, SNS determines it by validating the street address in PREMIS. Once the WC is known, SNS removes the loop reservation in LFACS, based on the reservation key used to initially build the reservation. If NUMLP is not provided in the old ACL, LFACS assumes that the entire reservation is to be removed. If the NUMLP provided in the old ACL exceeds the actual number of loops reserved in LFACS, LFACS removes the reservations on those found. (In LFACS 18.6, however, none of the loop reservations are removed). It should also be noted that LFACS removes reservations on the loops most *recently* reserved. If the NOTICE indicator is set, LFACS sends a confirmation notice to impacted organizations.

As a reminder:

The *ROSP section is one section in the UPDRSV request message. The structure of the sections in the UPDRSV request is as follows:

*C1 section
*PLHDR section
*SNSHDR section
*ROSP section.

The following data items may appear in the *ROSP section:

```
*ROSP{

  SVCLOC{                     Service location
     ACL{                     Action line
        SAGA=...;             SAG area† [conditional — required if ZIP not
                              specified]
        ZIP=...;              Zip code† [conditional — required if SAGA not
                              specified]

        ADDR{                  Address
           BADR{               Basic address
                BAD=...;       Basic address designation [opt]
                AHN=...;       Assigned house number [opt]
                STR=...;       Street name or descriptive address
                CNA=...;       Community name† [opt]
                STN=...;       State name† [opt]
           }
           SUPL{               Supplemental address
                SID=...;       Structure ID [opt]
                STYP=...;      Structure type [opt]
                EID=...;       Elevation ID [opt]
                ETYP=...;      Elevation type [opt]
                UID=...;       Unit ID [opt]
                UTYP=...;      Unit type [opt]
           }
        }
     }
  }

  CTL{                        Control data
     CTC=R;                   Control code
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
  }

  ACL{                      Old action line
     ACT = O;               Action code
     TN{                    Telephone number [conditional – required if CLS,
                            PFO, or SO not specified]
         NPA = ...;         NPA
         NXX = ...;         NXX
         LINE = ...;        Line
     }
     CLS = ...;             CLCI serial number [conditional – required if TN,
                            PFO, or SO not specified]
     DPA = ...;             Different premises address identifier [opt]
     CKL = ...;             Circuit location identifier [opt]
     PFO = ...;             Planned facilities order [conditional – required if
                            TN, CLS, or SO not specified]
     SO = ...;              Service order [conditional – required if TN, CLS,
                            or PFO not specified]
     NUMLP = ...;           Number of loops [opt]
  }
}%
```

4.6.4.1 SNS Processing of UPDRSV for Remove

Upon receiving a UPDRSV request, SNS generates an APPVAL request to PREMIS to SAG-validate the input address, as described in the INQSAG contract (unless the WC is provided in the input request). If the PREMIS SAG validation is successful, SNS uses the PREMIS primary street address to construct the living unit address. Otherwise, SNS uses the input street address. In either case, SNS uses the input supplemental address data (LOC) to construct the living unit address. SNS then generates the following UPDRES request to LFACS for this living unit:

- Remove a loop reservation

  SNS generates a request to LFACS to remove a reservation established previously on one or more loops.

4.6.5 *ROSP Section Format: Change Loop Reservation (UPDRSV)

SNS uses the information provided in the *ROSP section of the UPDRSV request message to change a loop reservation established earlier in LFACS. If the WC is not specified in the input request, SNS determines it by validating the input street address in PREMIS. Once the WC is known, SNS changes the loop reservation in LFACS, based on the reservation key used to initially build the reservation. SNS requires enough information about the original reservation so that LFACS can remove it (in the same manner as in removing a loop reservation) and then establish the new one (in the same manner as in building a loop reservation). If NUMLP is not provided in the old ACL, LFACS assumes that the entire reservation is to be removed. If the NUMLP provided in the old ACL exceeds the actual number of loops reserved in LFACS, LFACS removes the reservations on those found. (In LFACS 18.6, however, none of the loop reservations are removed). It should also be noted that LFACS removes reservations on loops most *recently* reserved. If the NOTICE indicator is set, LFACS sends a confirmation notice to impacted organizations.

As a reminder:

The *ROSP section is one section in the UPDRSV request message. The structure of the sections in the UPDRSV request is as follows:

*C1 section
*PLHDR section
*SNSHDR section
*ROSP section.

The following data items may appear in the *ROSP section:

```
*ROSP{

  SVCLOC{                      Service location
     ACL{                      Action line
        SAGA=...;              SAG area† [conditional – required if ZIP not
                               specified]
        ZIP=...;               Zip code† [conditional – required if SAGA not
                               specified]
        ADDR{                  Address
           BADR{               Basic address
                BAD=...;       Basic address designation [opt]
                AHN=...;       Assigned house number [opt]
                STR=...;       Street name or descriptive address
                CNA=...;       Community name† [opt]
                STN=...;       State name† [opt]
           }
           SUPL{               Supplemental address
                SID=...;       Structure ID [opt]
                STYP=...;      Structure type [opt]
                EID=...;       Elevation ID [opt]
                ETYP=...;      Elevation type [opt]
                UID=...;       Unit ID [opt]
                UTYP=...;      Unit type [opt]
           }
        }
     }
  }

  CTL{                        Control data
     CTC=C;                   Control code
  }
```

† The *first* time an address is validated, either SAGA *or* ZIP is required, and CNA and STN are optional. Once the address is determined to be valid, all subsequent contracts involving this address *should* contain the CNA, STN, SAGA, and ZIP obtained from the initial validation, along with the primary address information.

```
ACL{                     Old action line
    ACT=O;               Action code
    TN{                  Telephone number [conditional – required if CLS, PFO,
                         or SO not specified]
         NPA=...;        NPA
         NXX=...;        NXX
         LINE=...;       Line
    }
    CLS=...;             CLCI serial number [conditional – required if TN, PFO,
                         or SO not specified]
    DPA=...;             Different premises address identifier [opt]
    CKL=...;             Circuit location identifier [opt]
    PFO=...;             Planned facilities order [conditional – required if TN,
                         CLS, or SO not specified]
    SO=...;              Service order [conditional – required if TN, CLS, or PFO
                         not specified]
    NUMLP=...;           Number of loops [opt]
    WC=...;              Wire center [opt]
    NOTICE=...;          Confirmation notice indicator [opt]
}
ACL{                     New action line
    ACT=N                Action code
    PROD=...;            Product name
    MADN=...;            MADN flag [opt] <Y>
    FO=...;              Foreign office indicator [opt] <Y>
    NUMLP=...;           Number of loops [conditional – required when PFO or
                         SO specified]
    ACCT=...;            Customer account [opt]
    MKSEG=...;           Market segment indicator [opt] <R, P, M, C, G, I, or A>
    TN{                  Telephone number [conditional – required if CLS, PFO,
                         or SO not specified]
         NPA=...;        NPA
         NXX=...;        NXX
         LINE=...;       Line
    }
    CLS=...;             CLCI serial number [conditional – required if TN, PFO,
                         or SO not specified]
    DPA=...;             Different premises address identifier [opt]
    CKL=...;             Circuit location identifier [opt]
```

```
        PFO=...;              Planned facilities order [conditional – required if TN,
                              CLS, or SO not specified]
        SO=...;               Service order [conditional – required if TN, CLS, or PFO
                              not specified]
        SIG=...;              Signaling OEC [opt – default = L] <L, G or B>
        RSVDATE{              Reservation due date
            YEAR=...;         Year
            MONTH=...;        Month
            DAY=...;          Day
        }
        RSVRMK=...;           Reservation remark [opt]
    }
}%
```

4.6.5.1 SNS Processing of UPDRSV for Change

Upon receiving a UPDRSV request, SNS generates an APPVAL request to PREMIS to SAG-validate the input address, as described in the INQSAG contract (unless the WC is provided in the input request). If the PREMIS SAG validation is successful, SNS uses the PREMIS primary street address to construct the living unit address. Otherwise, SNS uses the input street address. In either case, SNS uses the input supplemental address data (LOC) to construct the living unit address. SNS then generates the following UPDRES request to LFACS for this living unit:

- Change a loop reservation

  SNS generates a request to LFACS to remove a loop reservation established previously and establish a new reservation (as described above).

4.7 Outside Plant Response (*OSPR) Section

The *OSPR section returned in the response message indicates the results of the INQLAI or UPDRSV contracts. It may contain address and network-related information, as well as identify errors and discrepancies.

4.7.1 General *OSPR Section Format

The data items† for the *OSPR section are shown in the following general format. The items shown in regular typeface reflect the data provided in the original request (*ROSP section). The original input data from the *ROSP section of the request message is always provided in the corresponding *OSPR section of the response message. (The original input data shown in this general format does not indicate multiple occurrences of data items, since that depends on the information provided in the request message.) The items shown in bold typeface represent the response data that SNS supplies. The first group of response data – from IERC through LPRC – includes the CTL aggregate return codes. The second group of response data – from SAGAM through LPINFO – shows the additional data aggregates that may appear in the *OSPR section.‡ The return codes and the additional data aggregates are explained in the sections that follow the general format.

> As a reminder:
>
> The *OSPR section is one section in the response message. The structure of the sections in the response message is as follows:
>
> *C2 section
> *PLHDR section
> *SNSHDR section
> *OSPR section.

† Data items shown in the *general* section format do not indicate whether they are optional, conditional or required. This information is supplied in the subsequent sections.

‡ In an actual *OSPR section, each of these additional data aggregates would have data items under it. These data items are not shown in the general format; however, they are explained in subsequent sections.

```
*OSPR{
  SVCLOC{                       Service location
     ACL{                       Action line
         SAGA=...;              SAG area
         ZIP=...;               Zip code
         ADDR{                  Address
              BADR{             Basic address
                BAD=...;        Basic address designation
                AHN=...;        Assigned house number
                STR=...;        Street name
                CNA=...;        Community name
                STN=...;        State name
              }
              SUPL{             Supplemental address
                SID=...;        Structure ID
                STYP=...;       Structure type
                EID=...;        Elevation ID
                ETYP=...;       Elevation type
                UID=...;        Unit ID
                UTYP=...;       Unit type
              }
         }
     }
  }

  CTL{                          Control data
     CTC=...;                   Control code

     IERC=...;                  Input error return code {0-N}, where N = the
                                number of input errors detected
     SARC=...;                  Street address return code {0-1}
     LURC=...;                  Living unit return code {0-1}
     LPRC=...;                  Loop return code {0-1}
  }

  ACL{                          Action line {1-2}
     ACT=...;                   Action code
     PROD=...;                  Product name
     MADN=...;                  MADN flag
```

```
    FO=...;                  Foreign office indicator [opt] <Y>
    WC=...;                  Wire center
    NUMLP=...;               Number of loops
    ACCT=...;                Customer account
    MKSEG=...;               Market segment indicator
    TN{                      Telephone number
        NPA=...;             NPA
        NXX=...;             NXX
        LINE=...;            Line
    }
    CLS=...;                 CLCI serial number
    DPA=...;                 Different premises address identifier
    CKL=...;                 Circuit location identifier
    PFO=...;                 Planned facilities order
    SO=...;                  Service order
    SIG=...;                 Signaling OEC [opt - default = L] <L, G or B>
    RSVDATE{                 Reservation due date
        YEAR=...;            Year
        MONTH=...;           Month
        DAY=...;             Day
    }
    RSVRMK=...;              Reservation remark
    NOTICE=...;              Conference notice indicator
}

SAGAM{                       SAGA menu {0-1}
}
STRM{                        Street name menu {0-1}
}
DESCRIPM{                    Descriptive name menu {0-1}
}
ARGM{                        Address range menu {0-1}
}
ADDRM{                       Address menu {0-1}
}
STRARGM{                     Street name address range menu {0-1}
}
BADRM{                       Basic address menu {0-1}
}
```

```
  PARDESCRIP{               Descriptive partial match {0-1}
  }
  SIMSTRPT{                 Similar street report {0-1}
  }
  SAGINFO{                  SAG information {0-1}
  }
  STRINFO{                  Street address information {0-1}
  }
  LUM{                      Living unit menu {0-1}
  }
  LUINFO{                   Living unit information {0-1}
  }
  LPINFO{                   Loop information {0-1}
  }
}%
```

4.7.2 CTL Aggregate Return Codes

There are four return codes that may be included with the CTL aggregate data. In the general format above, they are shown in bold typeface under the CTL aggregate. These return codes are:

- *Input Error Return Code:*

  The contract invoker is responsible for providing SNS with a valid Outside Plant request message that conforms to the specifications outlined in this document. However, if SNS detects a critical *input error* condition that prevents it from continuing its processing, it notifies the contract invoker of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors.

- *Street Address Return Code:*

  The SARC tag indicates the validity of the street address provided on input. It is produced whenever the Outside Plant contract involves a request to PREMIS to SAG-validate the input street address.

- *Living Unit Return Code:*

  The LURC tag indicates the validity of the living unit address sent to LFACS. It is produced whenever the Outside Plant contract involves a request to LFACS.

- *Loop Return Code:*

  The LPRC tag indicates the results of the loop inquiry or loop reservation.

Refer to Appendix D for a description of each return code. (Some return codes are not applicable to all loop activities).

4.7.3 Additional Data Aggregates Provided in *OSPR Section

There are fourteen additional data aggregates that may appear in the *OSPR section. In the general format shown previously, they appear in bold typeface at the end of the *OSPR section. These additional data aggregates are included only when specific return codes and values appear in the CTL aggregate. Table 4-3 shows the relationships between return code values and additional data aggregates. The individual data items that would appear with the additional data aggregates are described in the sections that follow the table.

Table 4-3. *OSPR CTL Return Codes and Related Data Aggregates

Return Code	Value	Related Data Aggregate
SARC	010	SAGAM
SARC	004	STRM
SARC	006	DESCRIPM
SARC	007,014	ARGM
SARC	008	ADDRM
SARC	009	STRARGM
SARC	012,013	BADRM
SARC	013	PARDESCRIP
SARC	> 100	SAGINFO
SARC	102-106	STRINFO
LURC	004	SIMSTRPT
LURC	003	LUM
LURC	002	LUINFO
LPRC	101,102, 201,202	LPINFO

4.7.3.1 *OSPR Section Format: SAG Area Menu

The data items in the SAGAM aggregate of the *OSPR section are identical to those in the SAGAM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.2 *OSPR Section Format: Street Name Menu for More than One Partial Match

The data items in the STRM aggregate of the *OSPR section are identical to those in the STRM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.3 *OSPR Section Format: Descriptive Name Menu for More than One Partial Match

The data items in the DESCRIPM aggregate of the *OSPR section are identical to those in the DESCRIPM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.4 *OSPR Section Format: House Number/AHN Number Menu

The data items in the ARGM aggregate of the *OSPR section are identical to those in the ARGM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.5 *OSPR Section Format: Address Duplicated (Community/State Menu)

The data items in the ADDRM aggregate of the *OSPR section are identical to those in the ADDRM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.6 *OSPR Section Format: Street Name/Address Range Menu

The data items in the STRARGM aggregate of the *OSPR section are identical to those in the STRARGM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.7 *OSPR Section Format: Select Basic Address Menu

The data items in the BADRM aggregate of the *OSPR section are identical to those in the BADRM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.8 *OSPR Section Format: Partial Match on Descriptive

The data items in the PARDESCRIP aggregate of the *OSPR section are identical to those in the PARDESCRIP aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.9 *OSPR Section Format: Living Unit Menu

The data items in the LUM aggregate of the *OSPR section are identical to those in the LUM aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.10 *OSPR Section Format: Similar Street Report

The data items in the SIMSTRPT aggregate of the *OSPR section are identical to those in the SIMSTRPT aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.11 *OSPR Section Format: SAG Information

The following identifies the *OSPR SAG-related data items returned for a valid input street address. A SAGINFO aggregate is returned for any of the following return codes in the CTL aggregate:

SARC = all return codes greater than 100.

```
SAGINFO{                    SAG information
   SAGA=...;                SAG area
   CNA1=...;                Alternate community name 1 [opt]
   CNA2=...;                Alternate community name 2 [opt]
   ALT=...;                 Alternate address [opt]
   DESCRIP=...;             Descriptive address [opt]
   WC=...;                  Wire center
   ZIP=...;                 Zip code [opt]
   CNA=...;                 Community name
   CNARQ=...;               Community name required indicator [opt]
   STN=...;                 State name
   STNRQ=...;               State name required indicator [opt]
}
```

4.7.3.12 *OSPR Section Format: Modified Street Information

The data items in the STRINFO aggregate of the *OSPR section are identical to those in the STRINFO aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.13 *OSPR Section Format: Modified Living Unit Information

The data items in the LUINFO aggregate of the *OSPR section are identical to those in the LUINFO aggregate of the *SNSR section. (See the description of the *SNSR section in this document.)

4.7.3.14 *OSPR Section Format: Successful Loop Inquiry

The following identifies the *OSPR data items returned when the loop inquiry is partially successful. This type of *OSPR is returned for any of the following return codes in the CTL aggregate:

LPRC = 101 and 102.

```
LPINFO{                     Loop information
   NUMSP=...;               Number of spare loops
   NUMCF=...;               Number of connected facilities
}
```

4.7.3.15 *OSPR Section Format: Successful Loop Reservation

The following identifies the *OSPR data items returned when the loop reservation is partially successful. This type of *OSPR is returned for any of the following return codes in the CTL aggregate:

LPRC = 201 and 202.

```
LPINFO{                     Loop information
   NUMRSV=...;              Number of reserved loops
   NUMCF=...;               Number of connected facilities
}
```

4.8 Request for TN List Selection/Return (*RTNL) Section

The *RTNL section is required in the request message of the TN List Selection/Return contract (UPDTNL). It contains the data needed to *select* or *return* TNs from a TN list established for a regular Centrex group.

4.8.1 General *RTNL Section Format

The data items† for the *RTNL section are:

```
*RTNL{
  CTL{                       Control data
     TNACT=...;              TN activity
  }

  ACL{                       Action line
     MAINTN{                 Main TN
        NPA=...;             Main NPA
        NXX=...;             Main NXX
        LINE=...;            Main line
     }
     SPTN{                   Specified TN {0-1}
        TN{                  Telephone number
          NPA=...;           NPA
          NXX=...;           NXX
          LINE=...;          Line
        }
        TNRNG{               TN range
          NPA=...;           NPA
          NXX=...;           NXX
          LOLN=...;          Low line
          HILN=...;          High line
        }
     }
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
    UNSPTN{                 Unspecified TN {0-1}
      NUMTN=...;            Number of TNs
      CON=...;              Consecutive indicator <Y=Yes>
      NPA=...;              NPA
      NXX=...;              NXX
    }
  }
}%
```

The CTL aggregate identifies certain control information. In particular, the TN activity code (TNACT) is used to determine the type of activity to be performed:

- Select TNs (TNACT=S)
- Return TNs (TNACT=R).

The ACL aggregate contains the specific information required to perform the TN list activity. The information that SNS requires in each ACL aggregate varies, depending upon the TNACT code that appears in the CTL aggregate. The following subsections explain how the *RTNL section should be constructed.

4.8.2 *RTNL Section Format: Selection of Non-Specific TNs

SNS uses the information provided in the *RTNL section to select non-specific TNs associated with a regular Centrex group. For an existing Centrex customer, a main TN (MAINTN) should be specified on input so that SNS can *derive* the Centrex group to which it belongs. In the TNLIST database, each regular Centrex group is associated with a main TN. If SNS does not find that the MAINTN is a main TN associated with a Centrex group in TNLIST, SNS instead derives the group based on its existence in a TN list.

When a number of TNs is requested for selection, the *RTNL may optionally indicate whether the TNs should be selected from a particular NPA-NXX and/or whether the selected TNs should be consecutive.

> As a reminder:
>
> The *RTNL section is one section in the UPDTNL request message. The structure of the sections in the UPDTNL request is as follows:
>
> *C1 section
> *PLHDR section
> *SNSHDR section
> *RTNL section.

The following data items may appear in the *RTNL section:

```
*RTNL{

  CTL{                         Control data
     TNACT=S;                  TN activity
  }

  ACL{                         Action line
     MAINTN{                   Main TN
        NPA=...;               Main NPA
        NXX=...;               Main NXX
        LINE=...;              Main line
     }
     UNSPTN{                   Unspecified TN
        NUMTN=...;             Number of TNs to select
        CON=...;               Consecutive indicator <Y=Yes> [opt]
        NPA=...;               NPA [conditional - required if NXX specified]
        NXX=...;               NXX [conditional - required if NPA specified]
     }
  }
}%
```

4.8.3 *RTNL Section Format: TN Return

SNS uses the information provided in the *RTNL section to return TNs back to the original list from which they were selected. The *RTNL may identify these TNs as individual TNs or as ranges of TNs. Return on a range basis is only meant to minimize the amount of data provided in the input request.

> As a reminder:
>
> The *RTNL section is one section in the UPDTNL request message. The structure of the sections in the UPDTNL request is as follows:
>
> *C1 section
> *PLHDR section
> *SNSHDR section
> *RTNL section.

The following data items may appear in the *RTNL section:

```
*RTNL{

  CTL{                          Control data
     TNACT=R;                   TN activity
  }

  ACL{                          Action line
     SPTN{                      Specified TN
        TN{                     Telephone number {0-N}, where N=the number
                                of individual TNs to return
           NPA=...;             NPA
           NXX=...;             NXX
           LINE=...;            Line
        }
        TNRNG{                  TN range {0-N}, where N=the number of TN
                                ranges to return
           NPA=...;             NPA
           NXX=...;             NXX
           LOLN=...;            Low line
           HILN=...;            High line
        }
     }
  }
}%
```

4.9 TN List Selection/Return Response (*TNLR) Section

The *TNLR section returned in the response message indicates the results of the TN selection or return activity in the UPDTNL request message. It may also identify errors and discrepancies.

4.9.1 General *TNLR Section Format

The data items† for the *TNLR section are shown in the following general format. The items shown in regular typeface reflect the data provided in the original request (*RTNL section). The original input data from the *RTNL section of the request message is always provided in the corresponding *TNLR section of the response message. (The original input data shown in this general format does not indicate multiple occurrences of data items, since that depends on the information provided in the request message.) The items shown in bold typeface represent the response data that SNS supplies. The first group of response data – from IERC to TNRC – includes the CTL aggregate return codes. The second group of response data – from TNGRPINFO through SPTNINFO – shows the additional data aggregates that may appear in the *TNLR section.‡ The return codes and the additional data aggregates are explained in the sections that follow the general format.

As a reminder:

The *TNLR section is one section in the response message. The structure of the sections in the response message is as follows:

*C2 section
*PLHDR section
*SNSHDR section
*TNLR section.

† Data items shown in the *general* section format do not indicate whether they are optional, conditional or required. This information is supplied in the subsequent sections.

‡ In an actual *TNLR section, each of these additional data aggregates would have data items under it. These data items are not shown in the general format; however, they are explained in subsequent sections.

```
*TNLR{

  CTL{                          Control data
     TNACT=...;                 TN activity

     IERC=...;                  Input error return code {0-N}, where N = the
                                number of input errors detected
     TNRC=...;                  TN return code {0-1}
  }

  ACL{                          Action line
     MAINTN{                    Main TN
        NPA=...;                Main NPA
        NXX=...;                Main NXX
        LINE=...;               Main line
     }
     SPTN{                      Specified TN {0-1}
        TN{                     Telephone number
           NPA=...;             NPA
           NXX=...;             NXX
           LINE=...;            Line
        }
        TNRNG{                  TN range
           NPA=...;             NPA
           NXX=...;             NXX
           LOLN=...;            Low line
           HILN=...;            High line
        }
     }
     UNSPTN{                    Unspecified TN {0-1}
        NUMTN=...;              Number of TNs
        CON=...;                Consecutive indicator <Y=Yes>
        NPA=...;                NPA
        NXX=...;                NXX
     }
  }

  TNGRPINFO{                    TN group information {0-1}
  }
  UNSPTNINFO{                  Unspecified TN information {0-1}
  }
  SPTNINFO{                    Specified TN information {0-1}
  }
}%
```

4.9.2 CTL Aggregate Return Codes

There are two return codes that may be included with the CTL aggregate data. In the general format above, they are shown in bold typeface under the CTL aggregate. These return codes are:

- *Input Error Return Code:*

  The contract originator is responsible for providing SNS with a valid TN List Selection/Return request message that conforms to the specifications outlined in this document. However, if SNS detects a critical *input error* condition that prevents it from continuing its processing, it notifies the originator of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors.

- *TN Return Code:*

  The TNRC tag indicates the results of the TN selection or return.

Refer to Appendix D for a description of each return code. (Some return codes are not applicable to all TN activity types).

4.9.3 Additional Data Aggregates Provided in *TNLR Section

There are three additional data aggregates that may appear in the *TNLR section. In the general format shown previously, they appear in bold typeface at the end of the *TNLR section. These additional data aggregates are included only when specific values of the TNRC return code appear in the CTL aggregate. Table 4-4 shows the relationships between TNRC values and additional data aggregates. The individual data items that would appear with the additional data aggregates are described in the sections that follow the table.

Table 4-4. TNRC Values and Related Data Aggregates

TNRC Value	Related Data Aggregate
104-1, 105-1, 106-1	TNGRPINFO
104, 104-1, 106, 106-1, 107, 107-1, 108, 108-1	UNSPTNINFO
202	SPTNINFO

4.9.3.1 *TNLR Section Format: TN Group Information

The following identifies the *TNLR data items returned when the input MAINTN differs from the main TN of the regular Centrex group to which the input TN belongs.

A TNGRPINFO aggregate is returned for any of the following return codes in the CTL aggregate:

TNRC = 104-1, 105-1 and 106-1.

```
TNGRPINFO{              TN group information
   MAINTN{              Main TN
      NPA=...;          Main NPA
      NXX=...;          Main NXX
      LINE=...;         Main line
   }
}
```

4.9.3.2 *TNLR Section Format: TN Selection of Unspecified TNs

The following identifies the *TNLR data items returned to indicate the results of the TN selection when particular TNs were not specified. An UNSPTNINFO aggregate is returned for any of the following return codes in the CTL aggregate:

TNRC = 104, 104-1, 106, 106-1, 107, 107-1, 108, and 108-1.

```
UNSPTNINFO{             Unspecified TN information
   TNS{                 Successful TNs
      NUMSEL=...;       Number of TNs selected
      TN{               Telephone number {0-N}, where N = the number
                        of non-consecutive TNs selected
         NPA=...;       NPA
```

```
            NXX=...;            NXX
            LINE=...;           Line
        }
        TNRNG{                  TN range {0-N}, where N=the number of groups
                                of consecutive TNs selected
            NPA=...;            NPA
            NXX=...;            NXX
            LOLN=...;           Low line
            HILN=...;           High line
        }
    }
}
```

4.9.3.3 *TNLR Section Format: TN Return Information

The following identifies the *TNLR data items supplied when the TN return was only partially successful. An SPTNINFO aggregate is returned when the following return code appears in the CTL aggregate:

TNRC = 202.

```
SPTNINFO{                       Specified TN information
    TNS{                        Successful TNs
        NUMRET=...;             Number of TNs returned
        TN{                     Telephone number {0-N}, where N=the number
                                of individual specified TNs that were returned
            NPA=...;            NPA
            NXX=...;            NXX
            LINE=...;           Line
        }
        TNRNG{                  TN range {0-N}, where N=the number of
                                specified TNs that were returned, grouped in ranges
                                (these ranges may not match the ranges in the
                                original request)
            NPA=...;            NPA
            NXX=...;            NXX
            LOLN=...;           Low line
            HILN=...;           High line
        }
    }
    TNERR{                    TN error
        TN{                   Telephone number {0-N}, where N = the number
                              of individual TNs that could not be returned
            NPA = ...;        NPA
            NXX = ...;        NXX
            LINE = ...;       Line
        }
        TNRNG{                TN range {0-N}, where N = the number of TNs
                              that could not be returned, in terms of ranges
            NPA = ...;        NPA
            NXX = ...;        NXX
            LOLN = ...;       Low line
            HILN = ...;       High line
        }
    }
}
```

5. Negotiation Dialogue Contracts

This section presents the five negotiation dialogue contracts defined to support Digital Centrex negotiations in the SNS Release 2.0. These dialogue contracts are:

- Line Negotiation Support Contract (SNSNEG) – Assists in service request administration, network data inquiries, and TN selection and return
- Service Specification Contract (SVCSPC) – Identifies associated service orders issued during the negotiation dialogue
- Order Completion Contract (ORDCMP) – Processes the completed order and ends the negotiation dialogue
- Hold Contract (HOLDSR) – Places a negotiation dialogue on hold and saves current service request data
- Retrieve Contract (RTRVSR) – Retrieves service request data that was placed on hold.

This section describes the FCIF structure of the negotiation dialogue contracts and explains the FCIF sections that are specific to the negotiation dialogue contracts. The FCIF sections that are common to all SNS contracts – *C1, *C2, *PLHDR, *SNSHDR, and *UMSG – are presented in Section 3 in this document.

5.1 Line Negotiation Support Contract (SNSNEG)

The Line Negotiation Support contract supports the negotiation of Digital Centrex lines and groups involved in a service request. This contract consists of a request message and a response message. The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section
*RSRA	Request for Service Request Administration section (optional)
*RNDI	Request for Network Data Inquiry section (optional)
*RTNI	Request for TNLIST Inventory section (optional)

NOTE

The *SNSHDR is required in the request message since the contract invoker must supply the date and time. The use of this data is described later.

The FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section (optional)
*SNSHDR	SNS Header section
*SRAR	Service Request Administration Response section (optional)
*NDIR	Network Data Inquiry Response section (optional)
*TNIR	TNLIST Inventory Response section (optional)

5.1.1 Line Negotiation Support Contract: Request Message

This contract is designed so that multiple requests may be initiated in one contract invocation (request message). These requests are structured in terms of the following FCIF sections:

- The *RSRA section is used to trigger a Service Request Administration activity in SNS:
  - Establish a service request.
  - Modify a service request.
  - Cancel a service request.

  Based on the type of negotiation, SNS derives data from the service request (e.g., serving network element) that is used for all subsequent requests pertaining to this session.
- The *RNDI section is used to request one or more Network Data Inquiries.
- The *RTNI section is used to request one or more TNLIST Inventory Updates.

Once a service request has been established in SNS, network data inquiries and network resource updates may be requested for this negotiation. It is possible to "couple" them along with a service request administration activity, provided the service request is not being canceled. However, if the service request administration activity is unsuccessful, SNS will *not* attempt to process the other requests in the message, since these requests are dependent upon the service request data.

With one exception, SNS does not impose any restrictions on *when* SNSNEG contracts can be invoked. (This exception is described later.) The contract is designed to support asynchronous invocations. The input date/time stamp supplied in the *SNSHDR is used to detect service request timing problems that may arise on rare occasions. By comparing this date/time stamp to the date and time that the service request was established or last modified, SNS can determine whether to continue processing the contract against the current service request data. Any "pending" requests sent before the last service request administration activity will not be completed (in case the data they rely on has been modified). There is one situation that does, however, impose a restriction. The contract invoker is expected to *wait* for the response to a service request administration activity before submitting any subsequent requests. The invoker should wait for a confirmation that the update did in fact take place. Otherwise, it would be ambiguous what service request data was actually in effect at the time subsequent requests were processed.

When SNS receives multiple network inquiries and/or network resource update requests in a message, it processes each request independently. Only one response message is returned when all requests have been processed.

5.1.1.1 Request for Service Request Administration (*RSRA) Section

The *RSRA section contains the data needed to perform a service request administration activity. When the Line Negotiation Support contract is invoked, SNS processes this type of request first, if one exists in the request message.

5.1.1.1.1 General *RSRA Section Format

The data items† for the *RSRA section are:

```
*RSRA{                                Request for Service Request Administration
    CTL{                              Control data
        SROPT=...;                    Service request option
                                      <E=Establish,M=Modify,C=Cancel>
    }                                 End CTL aggregate
    ACL{                              Action line
        NEGACT=...;                   Negotiation activity <A=Add Line(s),
                                      C=Change existing service>
        MAINTN{                       Main Centrex TN
            NPA=...;                  NPA
            NXX=...;                  NXX
            LINE=...;                 Line number
        }                             End MAINTN aggregate
        CUST=...;                     Customer name
        SA{                           Service address
            SAGA=...;                 SAG area
            ZIP=...;                  ZIP code
            ADDR{                     Address
                BADR{                 Basic address
                    BAD=...;          Basic address designation
                    AHN=...;          Assigned house number
                    STR=...;          Street name
                    CNA=...;          Community name
                    STN=...;          State name
                }                     End BADR aggregate
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
                SUPL{             Supplemental address data (LOC)
                    STYP=...;     Structure type
                    SID=...;      Structure ID
                    ETYP=...;     Elevation type
                    EID=...;      Elevation ID
                    UTYP=...;     Unit type
                    UID=...;      Unit ID
                }                 End SUPL aggregate
            }                     End ADDR aggregate
        }                         End SA aggregate
        SERVICE=C;                Type of service - Digital Centrex
        PROD=...;                 Product
        CONFIRM=...;              Confirmation to continue modification and
                                  return selected inventory <Y=Yes>
    }                             End ACL Aggregate
}%                                End *RSRA section
```

The CTL aggregate identifies certain control information. In particular, the service request option (SROPT) is used to determine the type of activity to be performed:

- Establish a service request (SROPT = E)
- Modify a service request (SROPT = M)
- Cancel a service request (SROPT = C).

The ACL aggregate contains the specific information required to perform the service request administration activity. The information that SNS requires in each ACL aggregate varies, depending upon the SROPT code that appears in the CTL aggregate. The following subsections explain how the *RSRA section should be constructed.

5.1.1.1.2 *RSRA Format: Establish Service Request

This request establishes a service request to add Digital Centrex lines or to change an existing Digital Centrex customer's service (e.g., line features). SNS determines the network element associated with the customer's Centrex group, based on the main TN provided on input. For better efficiency, the main TN of the Centrex group should be specified. If, however, the main Centrex TN is unknown, a TN within the Centrex group may be provided instead.

- If successful, all subsequent requests for this service request are based on this serving network element and service request data (unless modified).

- If unsuccessful, SNS does not establish the service request, since it is unable to provide further assistance.

```
*RSRA{                      Request for Service Request Administration
    CTL{                    Control data
        SROPT=E;            Service request option - establish service
                            request
    }                       End CTL aggregate
    ACL{                    Action line
        NEGACT=...;         Negotiation activity <A=Add Line(s),
                            C=Change existing service>
        MAINTN{             Main Centrex TN
            NPA=...;        NPA
            NXX=...;        NXX
            LINE=...;       Line number
        }                   End MAINTN aggregate
        CUST=...;           Customer name [opt]
        SA{                 Service address
            SAGA=...;       SAG area [conditional - required if ZIP not
                            specified]
            ZIP=...;        ZIP code [conditional - required if SAGA
                            not specified]
            ADDR{           Service address
                BADR{       Basic address
                    BAD=...;    Basic address designation [conditional -
                                required if AHN not specified]
                    AHN=...;    Assigned house number [conditional -
                                required if BAD not specified]
                    STR=...;    Street name
                    CNA=...;    Community name [opt]
                    STN=...;    State name [opt]
                }               End BADR aggregate
                SUPL{           Supplemental address data (LOC) [opt]
                    STYP=...;   Structure type [opt]
                    SID=...;    Structure ID [opt]
                    ETYP=...;   Elevation type [opt]
                    EID=...;    Elevation ID [opt]
                    UTYP=...;   Unit type [opt]
                    UID=...;    Unit ID [opt]
                }                         End SUPL aggregate
            }                             End ADDR aggregate
        }                                 End SA aggregate
        SERVICE=C;                        Type of service - Digital Centrex
        PROD=...;                         Product [opt]
    }                                     End ACL aggregate
}%                                        End *RSRA section
```

5.1.1.1.3 *RSRA Format: Modify Service Request

This request modifies a service request established earlier in SNS.† The modification request should include the new service request data along with the original data that has not changed. The only difference in the FCIF message format is the use of the *modify* service request option (SROPT=M). The type of service and negotiation activity must remain the same. A change to this type of information requires a cancellation of the existing service request and the establishment of a brand new service request.

When SNS receives a service request modification, it performs the processing that was described in the original request. If the modification results in a different Centrex group and TNs have already been selected, one of the following scenarios occurs:

- SNS does not proceed with the modification until it receives a confirmation (CONFIRM=Y) that the selected TNs should be returned. When SNS receives this confirmation, it first returns the TNs and then proceeds with the modification.
- SNS does not proceed with the modification if it results in a different network element, and service orders have already been issued based on the original network element. In this case, the service orders should be canceled and a brand new service request should be established.

Whenever a modification is successful, SNS indicates what other impacts the modification *may* have had on the negotiation. In this way, SNS alerts the contract invoker that data provided earlier may no longer be pertinent. SNS provides a warning when these situations occur, thereby allowing the negotiator to reuse as much of the negotiation as possible.

† The service request to be modified is identified by the service request number in the *C1 header.

5.1.1.1.4 *RSRA Format: Cancel Service Request

This request cancels the service request in SNS. In doing so, SNS also returns any selected inventory.

```
*RSRA{                      Request for Service Request Administration
    CTL{                    Control data
        SROPT=C;            Service request option- cancel service request
    }                       End CTL aggregate
}%                          End *RSRA section
```

5.1.1.2 Request for Network Data Inquiry (*RNDI) Section

The *RNDI section contains the data needed to perform a network data inquiry to obtain:

- Customer system data
- Network element characteristics
- Feature package information
- Feature group information
- In-service view of a series completion hunt group
- Customer premises equipment information

5.1.1.2.1 General *RNDI Section Format

The data items† for the *RNDI section are:

```
*RNDI{                          Request for Network Data Inquiry
    REC{                        Record {1-N}, where N equals the number of
                                Network Data Inquiry Requests
        CTL{                    Control data
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
            NDIOPT=...;             Network data inquiry option:
                                    <CUST=Customer System Data,
                                      NE=Network Element Characteristics,
                                      FPKG=Feature Package,
                                      FGRP=Feature Group,
                                      SCH=Series Completion Hunt Group In-
                                    Service View,
                                      CPE=Customer Premises Equipment>
        }                           End CTL Aggregate
        ACL{                        Action line
            ALL=...;                All information  <Y=Yes>
            FGRP=...;               Feature group name
            FPKG=...;               Feature package name
            TN{                     Telephone number (lead or pilot)
                NPA=...;            NPA
                NXX=...;            NXX
                LINE=...;           Line number
            }                       End TN aggregate
            SET{                    CPE set
                CPEMAN=...;         CPE manufacturer
                CPEMOD=...;         CPE model
            }                       End CPE set aggregate
        }                           End ACL aggregate
    }                               End REC aggregate
}%                                  End *RNDI section
```

Each record aggregate within the *RNDI section corresponds to a specific request and contains the data needed to process that request. The NDIOPT tag indicates which Network Data Inquiry option is being invoked. Only one option may be specified per record. However, multiple requests (REC aggregates) are allowed. The following sections describe the appropriate input data for each type of network data inquiry request.

5.1.1.2.2 *RNDI Format: Customer System Data Request

This request obtains the default customer system data for a Centrex group.

```
*RNDI{                              Request for Network Data Inquiry
    REC{                            Record
        CTL{                        Control data
```

```
            NDIOPT=CUST;   Customer system data option
        }                  End CTL Aggregate
    }                      End REC aggregate
}%                         End *RNDI section
```

5.1.1.2.3 *RNDI Format: Network Element Characteristics Request

This request obtains the characteristics of the serving network element.

```
*RNDI{                   Request for Network Data Inquiry
    REC{                 Record
        CTL{             Control data
            NDIOPT=NE;   Network element characteristics option
        }                End CTL aggregate
    }                    End REC aggregate
}%                       End *RNDI section
```

5.1.1.2.4 *RNDI Format: Feature Package Information Request

This request obtains feature package information, based on the requested product.

One of the following options may be requested:

- Provide a "list" of feature packages that can be offered to the customer. The optional FPKG and ALL tags are not specified on input.
- Obtain a description of a particular feature package. SNS identifies features associated with the specified feature package. The optional FPKG tag is specified on input.
- Provide *all* of the above feature package information. For each feature package available for the product, SNS identifies the features associated with each feature package. The optional ALL tag is specified on input.

```
*RNDI{                     Request for Network Data Inquiry
    REC{                   Record {1-N}, where N=the number of feature
                           package requests
        CTL{               Control data
            NDIOPT=FPKG;   Feature package option
        }                  End CTL aggregate
        ACL{               Action line
```

```
          FPKG=...;              Feature package name [opt]
          ALL=Y;                 All feature package information [opt]
      }                          End ACL aggregate
   }                             End REC aggregate
}%                               End *RNDI section
```

5.1.1.2.5 *RNDI Format: Feature Group Information Request

This request performs one of two functions:

1. Identify the feature groups that can be offered to the Digital Centrex customer, based on the customer's serving network element type and generic. The optional FGRP tag is not specified on input.
2. Identify the sub-features associated with the specified feature group that are actually available in the customer's serving network element. The optional FGRP tag is specified on input.

```
*RNDI{                           Request for Network Data Inquiry
   REC{                          Record {1-N}, where N=the number of
                                 feature group requests
      CTL{                       Control data
          NDIOPT=FGRP;           Feature group option
      }                          End CTL aggregate
      ACL{                       Action line
          FGRP=...;              Feature group name [opt]
      }                          End ACL aggregate
   }                             End REC aggregate
}%                               End *RNDI section
```

5.1.1.2.6 *RNDI Format: Series Completion Hunt Group In-Service View Request

This request determines the members of a series completion hunt group.

```
*RNDI{                         Request for Network Data Inquiry
   REC{                        Record {1-N}, where N=the number of SCH
                               requests
      CTL{                     Control data
```

```
            NDIOPT = SCH;   Series completion hunt group in-service view
                            option
        }                   End CTL aggregate
        ACL{                Action line
            TN{             Lead telephone number
                NPA = ...;  NPA
                NXX = ...;  NXX
                LINE = ...; Line number
            }               End TN aggregate
        }                   End ACL aggregate
    }                       End REC aggregate
}%                          End *RNDI section
```

5.1.1.2.7 *RNDI Format: CPE Information Request

This request performs one of two functions:

1. Identify the customer premises equipment that is compatible with the customer's serving network element. The SET aggregate is not specified on input.
2. Describe a particular CPE. The SET aggregate is specified on input.

```
*RNDI{                          Request for Network Data Inquiry
    REC{                        Record {1-N}, where N = the number of
                                CPE requests
        CTL{                    Control data
            NDIOPT = CPE;       Customer premises equipment option
        }                       End CTL aggregate
        ACL{                    Action line
            SET{                CPE set [opt]
                CPEMAN = ...;   CPE manufacturer
                CPEMOD = ...;   CPE model
            }                   End SET aggregate
        }                       End ACL aggregate
    }                           End REC aggregate
}%                              End *RNDI section
```

5.1.1.3 Request for TN List Inventory (*RTNI) Section

The *RTNI section contains the data needed to perform one or more TNLIST inventory updates to select or return TNs.

5.1.1.3.1 General *RTNI Section Format

The data items† for the *RTNI section are:

```
*RTNI{                        Request for TNLIST Inventory
    REC{                      Record {1-N}, where N = the number of
                              TN list inventory requests
        CTL{                  Control data
            TNACT=...;        TN activity <S=Select,R=Return>
        }                     End CTL aggregate
        ACL{                  Action line
            SPTN{             Specified TN(s)
                TN{           Telephone number
                    NPA=...;  NPA
                    NXX=...;  NXX
                    LINE=...; Line number
                }             End TN aggregate
                TNRNG{        TN range
                    NPA=...;  NPA
                    NXX=...;  NXX
                    LOLN=...; Low line number
                    HILN=...; High line number
                }             End TNRNG aggregate
            }                 End SPTN aggregate
            UNSPTN{           Unspecified TN(s)
                NUMTN=...;    Number of TNs
                CON=...;      Consecutive indicator <Y=Yes>
                NPA=...;      NPA
                NXX=...;      NXX
                ALLTN=...;    All TNs required flag <N=No>
            }                 End UNSPTN aggregate
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RTNI section
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

Each REC aggregate corresponds to an individual TNLIST inventory request and contains the data needed to process that request.

The CTL aggregate within each record identifies certain control information. In particular, the TN activity code (TNACT) is used to determine the type of activity to be performed:

- Select TNs (TNACT = S)
- Return TNs (TNACT = R).

The ACL aggregate contains the specific information required to perform the TNLIST activity. The information that SNS requires in each ACL aggregate varies, depending upon the TNACT code that appears in the CTL aggregate. In the case of TN selection, *either* specific TNs *or* non-specific TNs may be requested in each REC aggregate. The following subsections explain how the *RTNI section should be constructed.

5.1.1.3.2 *RTNI Section Format: Selection of Non-Specific TNs

This request selects non-specific TNs for a customer. A request may optionally indicate whether the TNs should be selected from a particular NPA-NXX or the TNs should be consecutive.

If all of the requested TNs are not available for selection, SNS does not select any TNs for the request, *unless* all TNs are not required (ALLTN = N). In this case, SNS selects the smaller subset of TNs that are available to partially fulfill the request.

```
*RTNI{                          Request for TNLIST Inventory
    REC{                        Record {1-N}, where N = the number of
                                TNlIST inventory requests
        CTL{                    Control data
            TNACT = S;          TN selection activity
        }                       End CTL aggregate
        ACL{                    Action line
            UNSPTN{             Unspecified TN(s)
                NUMTN = ...;    Number of TNs
                CON = ...;      Consecutive indicator < Y = Yes > [opt]
                NPA = ...;      NPA [opt - required if NXX specified]
                NXX = ...;      NXX [opt - required if NPA specified]
                ALLTN = ...;    All TNs required flag < N = No > [opt]
            }                   End UNSPTN aggregate
        }                         End ACL aggregate
    }                             End REC aggregate
}%                                End *RTNI section
```

5.1.1.3.3 *RTNI Section Format: Selection of Specific TNs

This request selects specific TNs for a customer. These TNs may be identified as individual TNs or as ranges of TNs. Selection on a range basis is meant to minimize the amount of data provided in the input request. (It has no bearing on the TN selection algorithm.)

If all of the requested TNs are not available for selection, SNS selects those TNs that can be selected.

```
*RTNI{                            Request for TNLIST Inventory
    REC{                          Record {1-N}, where N = the number of
                                  TNLIST inventory requests
        CTL{                      Control data
            TNACT = S;            TN selection activity
        }                         End CTL aggregate
        ACL{                      Action line
            SPTN{                 Specified TN(s)
                TN{               Telephone number {0-N}, where N = the
                                  number of individual TNs to select
                    NPA = ...;    NPA
                    NXX = ...;    NXX
                    LINE = ...;   Line number
                }                 End TN aggregate
                TNRNG{            TN range {0-N}, where N = the number of
                                  TN ranges to select
                    NPA = ...;    NPA
                    NXX = ...;    NXX
                    LOLN = ...;   Low line number
                    HILN = ...;   High line number
                }                 End TNRNG aggregate
            }                     End SPTN aggregate
        }                         End ACL aggregate
    }                             End REC aggregate
}%                                End *RTNI section
```

5.1.1.3.4 *RTNI Section Format: TN Return

This request returns TNs that were selected during the negotiation of this service request back to the original list from which they were selected. These TNs may be identified as individual TNs or as ranges of TNs. *SNS will not return any TNs that were not selected for this service request.* Return on a range basis is meant to minimize the amount of data provided in the input request.

```
*RTNI{                        Request for TNLIST Inventory
    REC{                      Record {1-N}, where N = the number of
                              TNLIST inventory requests
        CTL{                  Control data
            TNACT = R;        TN return activity
        }                     End CTL aggregate
        ACL{                  Action line
            SPTN{             Specified TN(s)
                TN{           Telephone number {0-N}, where N = the
                              number of individual TNs to return
                    NPA = ...;    NPA
                    NXX = ...;    NXX
                    LINE = ...;   Line number
                }             End TN aggregate
                TNRNG{        TN range {0-N}, where N = the number of
                              TN ranges to return
                    NPA = ...;    NPA
                    NXX = ...;    NXX
                    LOLN = ...;   Low line number
                    HILN = ...;   High line number
                }             End TNRNG aggregate
            }                 End SPTN aggregate
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RTNI section
```

5.1.2 Line Negotiation Support Contract: Response Message

The response message contains the results of each request. Similar to the request message, the response message is structured into FCIF sections. Each response message section corresponds to a section in the request message:

- The *SRAR section contains the results of the Service Request Administration activity in SNS.

- The *NDIR section contains the results of the requested Network Data Inquiry activities.
- The *TNIR section contains the results of the requested TNLIST Inventory Update activities.

5.1.2.1 Service Request Administration Response Section (*SRAR)

The *SRAR section contains the results of the service request administration activity.

5.1.2.1.1 General *SRAR Section Format

The data items† for the *SRAR section are shown in the following general format. The items shown in regular typeface reflect the data provided in the original request (*RSRA section). The original input data from the *RSRA section of the request message is always provided in the corresponding *SRAR section of the response message. (The original input data shown in this general format does not indicate multiple occurrences of data items, since that depends on the information provided in the request message.) The items shown in bold typeface represent the response data that SNS supplies. The first group of response data – IERC and SRRC – includes the CTL aggregate return codes. The second group of response data shows the additional data aggregates that may appear in the *SRAR section.‡ The return codes and the additional data aggregates are explained in the sections that follow the general format.

```
*SRAR{                         Service Request Administration Response
    CTL{                       Control data
        SROPT=...;             Service request option
                               <E=Establish,M=Modify,C=Cancel>
        IERC=...;              Input error return code {0-N}, where
                               N=the number of input errors
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional or required. This information is supplied in the subsequent sections.

‡ In an actual *SRAR section, each additional data aggregate would have data items under it. These data items are not shown in the general format; however, they are explained in subsequent sections.

```
        SRRC=...;               Service request return code {0-N}, where
                                N=the number of service request
                                processing conditions
    }                           End CTL aggregate
    ACL{                        Action line
        NEGACT=...;             Negotiation activity <A=Add Line(s),
                                C=Change existing service>
        MAINTN{                 Main Centrex TN
            NPA=...;            NPA
            NXX=...;            NXX
            LINE=...;           Line number
        }                       End MAINTN aggregate
        CUST=...;               Customer name
        SA{                     Service address
            SAGA=...;           SAG area
            ZIP=...;            ZIP code
            ADDR{               Address
                BADR{           Basic address
                    BAD=...;    Basic address designation
                    AHN=...;    Assigned house number
                    STR=...;    Street name
                    CNA=...;    Community name
                    STN=...;    State name
                }               End BADR aggregate
                SUPL{           Supplemental address data (LOC)
                    STYP=...;   Structure type
                    SID=...;    Structure ID
                    ETYP=...;   Elevation type
                    EID=...;    Elevation ID
                    UTYP=...;   Unit type
                    UID=...;    Unit ID
                }                 End SUPL aggregate
            }                     End ADDR aggregate
        }                         End SA aggregate
        SERVICE=C;                Type of service - Digital Centrex
        PROD=...;                 Product
        CONFIRM=...;              Confirmation to continue modification and
                                  return selected inventory <Y=Yes>
    }                             End ACL aggregate
    RETINV{                       Returned inventory
    }                             End RETINV aggregate
}%                                End *SRAR section
```

5.1.2.1.1.1 CTL Aggregate Return Codes

On a response, SNS may include various return codes in the CTL aggregate. Each provides more detail on any errors or special processing conditions encountered.

- *Input Error Return Code*

  The contract invoker is responsible for providing SNS with a valid request message that conforms to the specifications outlined in this document. However, if SNS detects a critical input error condition that prevents it from continuing its processing, it notifies the contract invoker of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times for multiple input errors.

- *Service Request Return Code*

  The SRRC tag indicates whether SNS was successful in performing the requested service request administration activity. It also alerts the contract invoker of any special processing conditions encountered (e.g., errors or warnings).

See Appendix D for a description of each return code.

If SNS was not successful in performing the requested Service Request Administration activity, it does *not* attempt to process any of the other requests (e.g., network data inquiries, network inventory updates) in the message.

There is one additional data aggregate that may appear in the *SRAR section. In the general format shown previously, it appears in bold typeface at the end of the *SRAR section. This additional data aggregate is included only when a particular value of the SRRC return code appears in the CTL aggregate, as described in the next section.

5.1.2.1.2 *SRAR Section Format: Returned Inventory

The RETINV aggregate identifies the selected inventory returned as a result of the service request modification or cancellation. A RETINV aggregate is supplied when the following return code appears in the CTL aggregate:

SRRC = 007.

```
RETINV{                     Returned Inventory
    TNI{                    TN inventory returned
        TN{                 Telephone number {0-N}, where N=the number
                            of individual TNs returned
            NPA=...;        NPA
            NXX=...;        NXX
            LINE=...;       Line number
        }                   End TN aggregate
        TNRNG{              TN range {0-N}, where N=the number of groups
                            of consecutive TNs returned
            NPA=...;        NPA
            NXX=...;        NXX
            LOLN=...;       Low line number
            HILN=...;       High line number
        }                   End TNRNG aggregate
    }                       End TNI aggregate
    TNERR{                  TN error - TN inventory that could not be returned
        TN{                 Telephone number {0-N}, where N=the number
                            of individual TNs that could not be returned
            NPA=...;        NPA
            NXX=...;        NXX
            LINE=...;       Line
        }                   End TN aggregate
        TNRNG{              TN range {0-N}, where N=the number of TNs
                            that could not be returned, in terms of ranges
            NPA=...;        NPA
            NXX=...;        NXX
            LOLN=...;       Low line
            HILN=...;       High line
        }                   End TNRNG aggregate
    }                       End TNERR aggregate
}                           End RETINV aggregate
```

5.1.2.2 Network Data Inquiry Response (*NDIR) Section

The *NDIR section contains the results of the network data inquiry(ies). It may contain network-related information, as well as identify errors and discrepancies. Each REC aggregate in the *NDIR section corresponds to a specific network data inquiry response.

5.1.2.2.1 General *NDIR Section Format

The data items[†] for the *NDIR section are shown in the following general format. The items shown in regular typeface reflect the data provided in the original request (*RNDI section). For each REC aggregate, the original input data from the *RNDI section of the request message is always provided in the corresponding *NDIR section of the response message. (The original input data shown in this general format does not indicate multiple occurrences of data items, since that depends on the information provided in the request message.) The items shown in bold typeface represent the response data that SNS supplies. The first group of response data – IERC and NDRC – includes the CTL aggregate return codes. The second group of response data – from CUSTINFO through CPEINFO – shows the additional data aggregates that may appear in the *NDIR section.[‡] The return codes and the additional data aggregates are explained in the sections that follow the general format.

```
*NDIR{                          Network Data Inquiry Response
    REC{                        Record {1-N}, where N equals the number
                                of Network Data Inquiry Responses
        CTL{                    Control data
```

[†] Data items shown in the *general* section format do not indicate whether they are optional, conditional or required. This information is supplied in the subsequent sections.

[‡] In an actual *NDIR section, each of these additional data aggregates would have data items under it. These data items are not shown in the general format; however, they are explained in subsequent sections.

```
          NDIOPT=...;          Network data inquiry option:
                               <CUST=Customer System Data,
                                 NE=Network Element Characteristics,
                                 FPKG=Feature Package,
                                 FGRP=Feature Group,
                                 SCH=Series Completion Hunt Group
                               In-
                                 Service View,
                                 CPE=Customer Premises Equipment>
          IERC=...;            Input error return code {0-N}, where
                               N=the number of input errors
          NDRC=...;            Network data return code {0-1}
     }                         End CTL aggregate
     ACL{                      Action line
          ALL=...;             All information <Y=Yes>
          FGRP=...;            Feature group name
          FPKG=...;            Feature package name
          SET{                 CPE set
               CPEMAN=...;     CPE manufacturer
               CPEMOD=...;     CPE model
          }                    End SET aggregate
          TN{                  Telephone number (lead or pilot)
               NPA=...;        NPA
               NXX=...;        NXX
               LINE=...;       Line number
          }                    End TN aggregate
     }                         End ACL aggregate
     CUSTINFO{                 Customer system information {0-1}
     }                         End CUSTINFO aggregate
     NEINFO{                   Network element information {0-1}
     }                         End NEINFO aggregate
```

```
         FPKGINFO{              Feature package information {0-1}
         }                      End FPKGINFO aggregate
         FGRPINFO{              Feature group information {0-1}
         }                      End FGRPINFO aggregate
         SCHINFO{               Series completion hunt group information
                                {0-1}
         }                      End SCHINFO aggregate
         CPEINFO{               Customer premises equipment
                                information {0-1}
         }                      End CPEINFO aggregate
     }                          End REC aggregate
}%                              End *NDIR section
```

5.1.2.2.1.1 CTL Aggregate Return Codes

There are two return codes that may be included with the CTL aggregate data. In the general format above, they are shown in bold typeface under the CTL aggregate. These return codes are:

- *Input Error Return Code*

  The contract invoker is responsible for providing SNS with a valid request message that conforms to the specifications outlined in this document. However, if SNS detects a critical input error condition that prevents it from continuing its processing, it notifies the contract invoker of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times for multiple input errors.

- *Network Data Return Code*

  The NDRC tag indicates whether SNS was successful in obtaining the requested network data.

See Appendix D for a description of each return code.

The only situation in which SNS does not return at least one of the above return codes is when the request was coupled with an unsuccessful service request administration activity.

5.1.2.2.2 Additional Data Aggregates in *NDIR Section

There are six additional data aggregates that may appear in the *NDIR section. In the general format shown previously, they appear in bold typeface at the end of the *NDIR section. These additional data aggregates are included only when specific values of the NDRC return code appear in the CTL aggregate. Table 5-1 shows the relationship between the NDRC return code values and the additional data aggregates. The individual data items that would appear with the additional data aggregates are described in the sections that follow the table.

Table 5-1. NDRC Return Codes and Related Data Aggregates

NDRC Value	Related Data Aggregate
003	CUSTINFO
001	NEINFO
007, 008, or 010	FPKGINFO
011 or 013	FGRPINFO
028	SCHINFO
021 or 023	CPEINFO

5.1.2.2.3 *NDIR Section Format: Customer System Information

The CUSTINFO aggregate contains the default customer system data for the Centrex group. A CUSTINFO aggregate is returned when the following return code appears in the CTL aggregate:

NDRC = 003.

```
CUSTINFO{                  Customer system information
    CTXGRP{                Centrex group
        CTX=...;           Centrex group ID
        CLLI=...;          Network element CLLI code
        CAT=...;           Customer access treatment [opt]
        INTCPT=...;        Intercept status [opt]
        LCC=...;           Line class code [opt]
        PIC=...;           Pre-designated inter-exchange carrier [opt]
        MAINTN{            Main Centrex TN [opt]
            NPA=...;       NPA
            NXX=...;       NXX
            LINE=...;      Line number
        }                  End MAINTN aggregate
    }                      End CTXGRP aggregate
}                          End CUSTINFO aggregate
```

A main TN is returned only if it differs from the main TN provided when the service request was established.

5.1.2.2.4 *NDIR Section Format: Network Element Characteristics

The NEINFO aggregate describes the characteristics of the customer's serving network element. A NEINFO aggregate is returned when the following return code appears in the CTL aggregate:

NDRC = 001.

```
NEINFO{                    Network element information
    NECHARS{               Network element characteristics
        CLLI=...;          Network element CLLI code
        WC=...;            Wire center
        NETYP=...;         Network element type
        NEGEN=...;         Network element generic
        MXSCH=...;         Maximum members in a SCH group [opt]
    }                      End NECHARS aggregate
}                          End NEINFO aggregate
```

5.1.2.2.5 *NDIR Section Format: Feature Package Information

The FPKGINFO aggregate contains feature package information for the requested product on the serving network element. Depending upon the request, SNS may return:

- A list of feature packages, *or*
- A description of a particular feature package, *or*
- A description of all feature packages.

In the latter two cases, SNS provides a description of each feature associated with the requested feature package(s). The following information is obtained from the Feature Catalog:

- Feature Availability

  SNS indicates whether the feature is optional in the feature package and whether it is available on a regular station set and/or electronic set.

- Feature Activation

  For each type of set that the feature can be used on, SNS indicates if the feature may be activated by dialing an access code and/or by depressing a button (key) on the set. This information is useful in configuring the set.

- Feature Scope

  For each feature that is available on an electronic business set (PSET), SNS also indicates the scope of the feature:

  - Business *Set* Feature: Applies to all directory numbers on the set. This includes DMS-100 *key* features that can be assigned to more than one vacant key on the set.
  - *Subset* Feature: Assigned to a subset of directory numbers on the set.
  - *Directory Number* Feature: Assigned on an individual directory number basis.
  - *Primary Number* Feature: Assigned to the set's primary number. The primary number of the set always appears on key one.

A FPKGINFO aggregate is returned when any of the following return codes appear in the CTL aggregate:

NDRC = 007, 008 or 010.

```
FPKGINFO{                      Feature package information
    NE{                        Network element
        CLLI=...;              Network element CLLI code
        PROD=...;              Product
        FTRPKG{                Feature package {1-N}, where N=the
                               number of feature packages returned
            FPKG=...;          Feature package name
            FEATURE{           Feature {0-N}, where N=the number of
                               features returned
                FTR=...;       Network element feature name <see
                               feature catalog>
                FTRDES=...;    Feature description
                OPT=...;       Optional feature flag <Y=Yes> [opt]
                RSET{          Regular (station) set feature [opt]
                    FAI=...;   Feature Activation Indicator
                               <D=Dial accessed,
                                N=Not dial accessed or buttonable>
                }              End RSET aggregate
                ESET{          Electronic set feature [opt for DMS-100
                               features]
                    FTRSCP=...;      Scope of feature on electronic set
                                     <B=(Business) set feature,
                                       S=Subset feature,
                                       D=Directory number feature,
                                       P=Primary number feature>
                    FAI=...;         Feature Activation Indicator
                                     <B=Buttonable,
                                       D=Dial accessed,
                                       E=Either dial accessed or buttonable,
                                       N=Not dial accessed or buttonable>
                }                    End ESET aggregate
            }                        End FEATURE aggregate
        }                            End FTRPKG aggregate
    }                                End NE aggregate
}                                    End FPKGINFO
```

5.1.2.2.6 *NDIR Section Format: Feature Group Information

The FGRPINFO aggregate contains either:

- The feature groups that are available for the type of service, based on the customer's serving network element type and generic.
- The sub-features available in the requested feature group. SNS also obtains a description of each feature from the Feature Catalog (as described in Feature Package Information).

A FGRPINFO aggregate is returned when any of the following return codes appear in the CTL aggregate:

NDRC = 011 or 013.

```
FGRPINFO{                            Feature group information
    NE{                              Network element
        CLLI=...;                    Network element CLLI code
        SERVICE=C;                   Type of service - Digital Centrex
        FGRP{                        Feature group {1-N}, where N=the
                                     number of feature groups returned
            FGRP=...;                Feature group name <see feature catalog>
            FGRPDES=...;             Feature group description [opt]
            FEATURE{                Feature {0-N}, where N = the number of
                                    features returned
                FTR = ...;          Network element feature name < see
                                    feature catalog>
                FTRDES = ...;       Feature description
                RSET{               Regular (station) set feature [opt]
                    FAI = ...;      Feature Activation Indicator
                                    < D = Dial accessed,
                                      N = Not dial accessed or buttonable>
                }                   End RSET aggregate
                ESET{               Electronic set feature [opt for DMS-100
                                    features]
                    FTRSCP = ...;   Scope of feature on electronic set
                                    < B = (Business) set feature,
                                      S = Subset feature,
                                      D = Directory number feature,
                                      P = Primary number feature>
                    FAI = ...;      Feature Activation Indicator
                                    < B = Buttonable,
                                      D = Dial accessed,
                                      E = Either dial accessed or buttonable,
                                      N = Not dial accessed or buttonable>
                }                   End ESET aggregate
            }                       End FEATURE aggregate
        }                           End FGRP aggregate
    }                               End NE aggregate
}                                   End FGRPINFO aggregate
```

5.1.2.2.7 *NDIR Section Format: Series Completion Hunt Group Information

The SCHINFO aggregate identifies the members in a series completion hunt group. *Telephone numbers will appear in hunt sequence.* A SCHINFO aggregate is returned when the following return code appears in the CTL aggregate:

NDRC = 028.

```
SCHINFO{                Series completion hunt group information
    HTYP = ...;         Hunting type < RG = Regular, CIR = Circular>
    TN{                    Telephone number {1-N}, where N=the number
                           of members in the hunt group
        NPA=...;           NPA
        NXX=...;           NXX
        LINE=...;          Line number
    }                      End TN aggregate
}                          End SCHINFO aggregate
```

5.1.2.2.8 *NDIR Section Format: Customer Premises Equipment Information

The CPEINFO aggregate contains either:

- The CPE that are compatible with the customer's serving network element.
- A description of the requested CPE.

A CPEINFO aggregate is returned when any of the following return codes appears in the CTL aggregate:

NDRC = 021 or 023.

```
CPEINFO{                        Customer premises equipment information
    CLLI=...;                   Network element CLLI code
    CPE{                        CPE {1-N}, where N=the number of CPE
                                returned
        SET{                    CPE set
            CPEMAN=...;         CPE manufacturer
            CPEMOD=...;         CPE model
        }                       End SET aggregate
        CPENAME=...;            CPE name [opt]
        CPEREG=...;             CPE registration number [opt]
        CHARS{                  CPE characteristics [opt]
            DIS=...;            Display flag <Y=Yes> [opt]
            ONETOUCH=...;       One touch flag <Y=Yes> [opt]
            MAXKEY=...;         Max number of keys <1-999> [opt]
            MAXCA=...;          Maximum number of call appearances <1-999>
                                [opt]
            HKEY=...;           Dedicated hold key flag <Y=Yes> [opt]
            RKEY=...;           Dedicated release key flag <Y=Yes> [opt]
        }                       End CHARS aggregate
    }                           End CPE aggregate
}                               End CPEINFO aggregate
```

5.1.2.3 TNLIST Inventory Response Section (*TNIR)

The *TNIR section returned in the response message contains the results of the TNLIST inventory request(s). Each REC aggregate corresponds to a TNLIST database inventory response. It may also identify errors and discrepancies.

5.1.2.3.1 General *TNIR Section Format

The data items† for the *TNIR section are shown in the following general format. The items shown in regular typeface reflect the data provided in the original request (*RTNI section). For each REC aggregate, the original input data from the *RTNI section of the request message is always provided in the corresponding *TNIR section of the response message. (The original input data shown in this general format does not indicate multiple occurrences of data items, since that depends on the information provided in the request message.) The items shown in bold typeface represent the response data that SNS supplies. The first group of response data – IERC and TNRC – includes the CTL aggregate return codes. The second group of response data – from UNSPTNINFO through SPTNINFO – shows the additional data aggregates that may appear in the *TNIR section.‡ The return codes and the additional data aggregates are explained in the sections that follow the general format.

```
*TNIR{                          TNLIST Inventory Response
    REC{                        Record {1-N}, where N=the number of
                                TNLIST inventory responses
        CTL{                    Control data
            TNACT=...;          TN activity <S=Select,R=Return>
            IERC=...;           Input error return code {0-N}, where N =
                                the number of input errors detected
            TNRC=...;           TN return code {0-1}
        }                       End CTL aggregate
        ACL{                    Action line
            SPTN{               Specified TN(s)
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional or required. This information is supplied in the subsequent sections.

‡ In an actual *TNIR section, each of these additional data aggregates would have data items under it. These data items are not shown in the general format; however, they are explained in subsequent sections.

```
                TN{                     Telephone number
                    NPA=...;            NPA
                    NXX=...;            NXX
                    LINE=...;           Line number
                }                       End TN aggregate
                TNRNG{                  TN range
                    NPA=...;            NPA
                    NXX=...;            NXX
                    LOLN=...;           Low line number
                    HILN=...;           High line number
                }                       End TNRNG aggregate
            }                           End SPTN aggregate
            UNSPTN{                     Unspecified TN(s)
                NUMTN=...;              Number of TNs
                CON=...;                Consecutive indicator <Y=Yes>
                NPA=...;                NPA
                NXX=...;                NXX
                ALLTN=...;              All TNs required flag <N=No>
            }                           End UNSPTN aggregate
        }                               End ACL aggregate
        UNSPTNINFO{ ...                 Unspecified TN information [opt]
        }                               End UNSPTNINFO aggregate
        SPTNINFO{ ...                   Specified TN information [opt]
        }                               End SPTNINFO aggregate
    }                                   End REC aggregate
}%                                      End *TNIR section
```

5.1.2.3.1.1 CTL Aggregate Return Codes

There are two return codes that may be included with the CTL aggregate data. In the general format above, they are shown in bold typeface under the CTL aggregate. These return codes are:

- *Input Error Return Code*

  The contract originator is responsible for providing SNS with a valid request message that conforms to the specifications outlined in this document. However, if SNS detects a critical input error condition that prevents it from continuing its processing, it notifies the originator of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors.

- *TN Return Code*

  The TNRC tag indicates the results of the TN selection or return.

See Appendix D for a description of each return code. (Some return codes are not applicable to all TN activity types).

The only situation in which SNS does not return at least one of the above return codes is when the request was coupled with an unsuccessful service request administration activity.

There are two additional data aggregates that may appear in the *TNIR section. In the general format shown previously, they appear in bold typeface at the end of the *TNIR section. These additional data aggregates are included only when specific values of the TNRC return code appear in the CTL aggregate.

5.1.2.3.2 *TNIR Section Format: TN Selection of Unspecified TNs

The following identifies the *TNIR data items returned to indicate the results of the TN selection when particular TNs were not specified in the request. An UNSPTNINFO aggregate is returned for any of the following return codes in the CTL aggregate:

TNRC = 104, 106, 107 and 108.

```
UNSPTNINFO{                 Unspecified TN information
    CLLI=...;               Network element CLLI code
    CTX=...;                Centrex group ID
    SERVICE=C;              Centrex service
    TNS{                    Successful TNs
        NUMSEL=...;         Number of TNs selected
        TN{                 Telephone number {0-N}, where N=the number
                            of non-consecutive TNs selected
            NPA=...;        NPA
            NXX=...;        NXX
            LINE=...;       Line
        }                   End TN aggregate
        TNRNG{              TN range {0-N}, where N=the number of groups
                            of consecutive TNs selected
            NPA=...;        NPA
            NXX=...;        NXX
            LOLN=...;       Low line
            HILN=...;       High line
        }                   End TNRNG aggregate
    }                       End TNS aggregate
}                           End UNSPTNINFO aggregate
```

5.1.2.3.3 *TNIR Section Format: TN Selection of Specific TNs

The following identifies the *TNIR data items returned to indicate the results of the TN selection. The following data items are provided when the selection of specified TNs was not entirely successful.

This type of SPTNINFO aggregate is returned for any of the following return codes in the CTL aggregate:

TNRC = 101, 102 and 103.

```
SPTNINFO{                   Specified TN information
    CLLI=...;               Network element CLLI code
    CTX=...;                Centrex group ID
    SERVICE=...;            Centrex service
    TNS{                    Successful TNs
        NUMSEL=...;         Number of TNs selected
        TN{                 Telephone number {0-N}, where N=the number
                            of individual specified TNs that were selected
            NPA=...;        NPA
            NXX=...;        NXX
            LINE=...;       Line
        }                   End TN aggregate
        TNRNG{              TN range {0-N}, where N=the number of
                            specified TNs that were selected, grouped in ranges
                            (these ranges may not match the ranges in the
                            original request)
            NPA=...;        NPA
            NXX=...;        NXX
            LOLN=...;       Low line
            HILN=...;       High line
        }                   End TNRNG aggregate
    }                       End TNS aggregate
    TNERR{                  TN error
        TN{                 Telephone number {0-N}, where N=the number
                            of specific TNs that could not be selected
            NPA=...;        NPA
            NXX=...;        NXX
            LINE=...;       Line number
            SELERR=...;     Selection error
                            < U=Unselectable,
                              K=Undesignated,
                              S=Selected,
                              I=Invalid>
        }                   End TN aggregate
    }                       End TNERR aggregate
}                           End SPTNINFO aggregate
```

When SNS is unable to select a specified TN, it identifies the reason for the failure via the SELERR tag:

- TN was not selected because it is marked *unselectable* in TNLIST (SELERR=U).
- TN was not selected because its status is *undesignated* in TNLIST (SELERR=K).
- TN was not selected because it is already marked as *selected* in TNLIST (SELERR=S).
- TN was not selected because it is *invalid* for the Centrex group (SELERR=I).

5.1.2.3.4 *TNIR Section Format: TN Return Information

The following identifies the *TNIR data items supplied when the TN return was only partially successful. A SPTNINFO aggregate is returned when the following return code appears in the CTL aggregate:

TNRC = 202.

```
SPTNINFO{                   Specified TN information
    TNS{                    Successful TNs
        NUMRET=...;         Number of TNs returned
        TN{                 Telephone number {0-N}, where N=the number
                            of individual specified TNs that were returned
            NPA=...;        NPA
            NXX=...;        NXX
            LINE=...;       Line
        }                   End TN aggregate
        TNRNG{                   TN range {0-N}, where N = the number of
                                 specified TNs that were returned, grouped in ranges
                                 (these ranges may not match the ranges in the
                                 original request)
            NPA = ...;           NPA
            NXX = ...;           NXX
            LOLN = ...;          Low line
            HILN = ...;          High line
        }                        End TNRNG aggregate
    }                            End TNS aggregate
    TNERR{                       TN error
        TN{                      Telephone number {0-N}, where N = the number
                                 of individual TNs that could not be returned
            NPA = ...;           NPA
            NXX = ...;           NXX
            LINE = ...;          Line
        }                        End TN aggregate
        TNRNG{                   TN range {0-N}, where N = the number of TNs
                                 that could not be returned, in terms of ranges
            NPA = ...;           NPA
            NXX = ...;           NXX
            LOLN = ...;          Low line
            HILN = ...;          High line
        }                        End TNRNG aggregate
    }                            End TNERR aggregate
}                                End SPTNINFO aggregate
```

5.2 Service Specification Contract (SVCSPC)

This abridged version of the Service Specification contract (SVCSPC) is used to identify those service orders that were issued during the negotiation dialogue. SNS uses this data to administer network resources and clean up the session when the corresponding Order Completion contract is invoked. The Service Specification contract should be invoked whenever a new order is issued for this service request. This contract consists of a request message and a response message.

- The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section [opt]
*SVCSPEC	Service specification section

- The FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section [opt]
*SNSHDR	SNS Header section
*SVCSPEC	Service specification section

5.2.1 Service Specification Contract Request Message

The service specification data provided in the request message consists of a minimal set of order identification data.

5.2.1.1 Service Specification (*SVCSPEC) Section Format

The *SVCSPEC FCIF section contains the service specification data for one or more orders associated with the service request. The FCIF presented below identifies the data that may be included in the *SVCSPEC section.

```
*SVCSPEC{                     Service specification
    REC{                      Record {1-N}, where N = the number of
                              orders
        ID{                   Identification
            ORDID{            Order identifier
```

```
                ORDTYP=C;       Order type: Change
                ORDNUM=...;     Order number
            }                   End ORDID aggregate
            ORDPASS=PRE;        Order pass: pre-completion
        }                       End ID aggregate
    }                           End REC aggregate
}%                              End *SVCSPEC section
```

Each REC aggregate identifies an individual order issued during the negotiation. Multiple REC aggregates are allowed for negotiations involving multiple orders.

5.2.2 Service Specification Contract Response Message

The response message contains the results of processing the Service Specification contract.

5.2.2.1 Service Specification (*SVCSPEC) Section Format

In a response message, the *SVCSPEC FCIF section contains the service specification data provided on input. These data items are identified below in regular typeface. Additionally, SNS may return the output data identified in bold typeface.

```
*SVCSPEC{                       Service specification
    REC{                        Record {1-N}, where N=the number of
                                orders
        CTL{ ...                Control data
            SSRC=...;           Service specification return code {0-N},
                                where N=the number of processing
                                conditions
        }                       End CTL aggregate
        ID{ ...                 Identification
            IDRC=...;           Identification return code {0-1}
            ORDID{              Order identifier
                ORDTYP=C;       Order type: Change
                ORDNUM=...;     Order number
            }                   End ORDID aggregate
            ORDPASS=PRE;        Order pass: pre-completion
        }                       End ID aggregate
    }                           End REC aggregate
```

}%　　　　End *SVCSPEC section

SNS supplements the response message with one or more return codes:

- The *Service Specification Return Code* (SSRC) returned in the CTL aggregate indicates whether SNS was successful in processing the request. The SSRC may also identify an error in the ID data aggregate. When an SSRC of 002 is returned, the ID aggregate should be checked for errors.
- The *Identification Return Code* (IDRC) provided in the ID aggregate indicates the order identification error.

See Appendix D for more information on the possible SSRC and IDRC return code values.

5.3 Order Completion Contract (ORDCMP)

The Order Completion contract signifies the completion or cancellation of an order and triggers order related updates in the TNLIST database. This contract should only be invoked once for each completed or canceled order. The order update information provided is assumed to be based on the latest version of the order. When the Order Completion contract is invoked, SNS updates the status of the order to show that it is no longer pending. If appropriate, the SCH in-service views are also updated.

SNS does *not* update the TNLIST database if any errors are encountered while processing the order. In this case, SNS identifies the errors and indicates which updates would have otherwise been successful.

The Order Completion contract may also result in negotiation dialogue clean-up activities. If all orders associated with the negotiation dialogue have been completed, SNS returns all unused TNs selected for this service request, unless an order update error occurred. If such an error occurs, SNS is not able to determine which TNs were unused and therefore which ones to return. SNS ends the negotiation dialogue by purging the session and its related data.

This contract consists of a request and an *optional* response message.

- The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section [opt]
*ORD	Order section
*SCH	Series Completion Hunt Group section [opt]
*TNL	TN Line section [opt]

NOTE

The Service Request Number (SRN) field in the *C1 header is *optional.* It should be left blank if unknown at the time the contract is invoked. SNS then determines the appropriate SRN, based on information received earlier when the related Service Specification contract was invoked.

- The FCIF sections in the optional response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section [opt]
*ORD	Order section [opt]
*SCH	Series Completion Hunt Group section [opt]
*TNL	TN Line section [opt]
*SELINV	Selected inventory section [opt]
*ORDINFO	Order status information section [opt]

NOTE

The ORDCMP response message is not returned unless an error occurs while processing one or more orders associated with the service request. This message is routed to the ORDERERR destination identified in the Output Handler Destination Table.

5.3.1 Order Completion Contract Request Message

The request message contains information pertaining to the completed or canceled order. The *ORD section is required in the request message to identify the order. The optional *SCH and *TNL sections contain the order related updates that impact the TNLIST database.

5.3.1.1 Order (*ORD) Section

The data items for the *ORD section are:

```
*ORD{                     Order
    ORDID{                Order identifier
        ORDTYP=C;         Order type: change
        ORDNUM=...;       Order number
        CORSFX=...;       Correction suffix [opt]
    }                     End ORDID aggregate
    ORDPASS...;           Order pass
                          <PCN=Post completion,
                            CAN=Cancellation>
}%                        End *ORD section
```

The *ORD section identifies the order and indicates whether the order completed successfully (ORDPASS = PCN) or was canceled (ORDPASS = CAN).

In the SNS 2.0 release, the Order Completion contract is assumed to be the source of order update information. The actual network changes that affect the TNLIST database are supplied in the optional *SCH and *TNL sections.

5.3.1.2 Series Completion Hunt Group (*SCH) Section

If SCH in-service views are kept in SNS, the *SCH section should be included in the request message to identify order-related updates.

The data items for the *SCH section are:

```
*SCH{                       Series completion hunt group
    REC{                    Record {1-N}, where N = the number of SCH in-
                            service view updates
        CTL{                Control data
            CTC = ...;      Control code  < B = Build,C = Change,R = Remove >
        }                   End CTL aggregate
        ACL{                Action line {1-2}
            ACT = ...;      Action code  < O = Old,N = New >
            HTYP = ...;     Hunting type  < RG = Regular, CIR = Circular >
                            [conditional; required when ACT = N]
            TN{             Telephone number {1-N}, where N = the number of
                            members in the hunt group
                NPA = ...;  NPA
                NXX = ...;  NXX
                LINE = ...; Line number
            }               End TN aggregate
        }                   End ACL aggregate
    }                       End REC aggregate
}%                          End *SCH section
```

Each REC aggregate contains an SCH in-service view update.

The CTL aggregate within each record contains the control code (CTC) used to determine the type of update to be performed:

- Build the in-service view of an SCH group (CTC = B).
- Change the in-service view of an SCH group (CTC = C).
- Remove the in-service view of an SCH group (CTC = R).

The ACL aggregate contains the specific information required to perform the SCH in-service view update. The action code within the ACL aggregate specifies the type of ACL, old or new. An *old* action code (ACT = O) identifies outward activity, whereas a *new* action code (ACT = N) indicates inward activity. The type of ACL aggregate required depends on the CTC code that appears in the CTL aggregate.

- A build request requires a new ACL.
- A change request requires both an old and a new ACL.
- A remove request requires an old ACL.

TN entries in the ACL aggregate must appear in hunt sequence.

5.3.1.3 TN Line (*TNL) Section

This FCIF section identifies TNs that are working TNs as a result of the order. SNS uses this information to compare against the selected TNs. Those TNs that were selected during the negotiation, but not used, are returned.

The data items for the *TNL section are:

```
*TNL{                       TN Line
    REC{                    Record {1-N}, where N = the number of TNL
                            requests
        CTL{                Control data
            CTC = B;        Control code: Build
        }                   End CTL aggregate
        ACL{                Action line
            ACT = N;        Action code: New
            TN{             Telephone number
                NPA = ...;  NPA
                NXX = ...;  NXX
                LINE = ...; Line number
            }               End TN aggregate
        }                   End ACL aggregate
    }                       End REC aggregate
}%                          End *TNL section
```

Each REC aggregate represents inward TN activity on the order. Multiple REC aggregates are allowed for multiple working TNs.

The CTL aggregate within each record contains the control code (CTC). In SNS 2.0, build (CTC = B) is the only control code supported.

The ACL aggregate contains a *new* action code (ACT = N) and identifies the new working TN.

5.3.2 Order Completion Contract Response Message

A response message is returned under one or both of the following circumstances:

- SNS encounters one or more errors while processing the order. In this situation, the *ORD, *SCH and *TNL sections provided on input are returned to identify the error (e.g., TNLIST discrepancy).
- SNS is unable to return unused network resources selected for the service request. When an order update error occurs, SNS does *not* automatically return unused network resources. It does, however, identify those resources that were selected. This information is provided in the *SELINV section when the last order associated with the negotiation has been completed or canceled. In this situation, an *ORDINFO section is also returned to identify those orders in the negotiation that could not be processed successfully in SNS.

  The *SELINV and *ORDINFO sections are not returned when there is no remaining selected inventory.

5.3.2.1 Order (*ORD) Section

The presence of the *ORD section in the response message indicates that SNS could not successfully process the order.

The data items for the *ORD section are identical to those provided on input and are identified below in regular typeface. In addition, SNS may optionally return the IERC return code shown in bold typeface.

```
*ORD{                        Order
    IERC=...;                Input error return code {0-N}
    ORDID{                   Order identifier
        ORDTYP=C;            Order type: change
        ORDNUM=...;          Order number
        CORSFX=...;          Correction suffix [opt]
    }                        End ORDID aggregate
```

```
          ORDPASS...;                 Order pass
                                      < PCN = Post completion,
                                        CAN = Cancellation >
}%                                    End *ORD section
```

The contract invoker is responsible for providing SNS with a valid request message that conforms to the specifications outlined in this document. However, if SNS detects a critical input error condition that prevents it from processing the order updates, it notifies the contract invoker of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times for multiple input errors.

When an input error condition is found in the *ORD section, SNS does *not* attempt to process the order updates in the *SCH or *TNL sections of the message.

5.3.2.2 Series Completion Hunt Group (*SCH) Section

When SNS is unsuccessful in processing the order, the *SCH section is provided in the response message, if received on input. Each REC aggregate within the *SCH section always contains the original input data. These data items are identified below in regular typeface. In addition, SNS may optionally return the SCHRC return code shown in bold typeface.

```
*SCH{                                 Series completion hunt group
     REC{                             Record {1-N}, where N = the number of SCH in-
                                      service view updates
          CTL{                        Control data
               CTC = ...;             Control code
                                      < B = Build,C = Change,R = Remove >
               SCHRC = ...;           SCH return code {0-N}
          }                           End CTL aggregate
          ACL{                        Action line {1-2}
               ACT = ...;             Action code  < O = Old, N = New >
               HTYP = ...;            Hunting type  < RG = Regular, CIR = Circular >
               TN{                    Telephone number {1-N}, where N = the number
                                      of members in the hunt group
                    NPA = ...;        NPA
                    NXX = ...;        NXX
                    LINE = ...;       Line number
               }                      End TN aggregate
```

```
        }                          End ACL aggregate
    }                              End REC aggregate
}%                                 End *SCH section
```

On a response, SNS may include a Series Completion Hunt Return Code. The SCHRC tag supplied in the CTL aggregate indicates whether SNS was "successful" in processing the SCH order update.

> **NOTE**
>
> As explained earlier, SNS does not perform partial order updates in the TNLIST database. As a result, the TNLIST database is *not* actually updated when an order error occurs, despite any "successful" return codes provided in the response message. The purpose of these return codes is to identify those order updates that would have been successful, had other errors not occurred.

See Appendix D for a description of the possible SCHRC return codes.

5.3.2.3 TN Line (*TNL) Section

When SNS is unsuccessful in processing the order, the *TNL section is provided in the response message, if received on input. Each REC aggregate within the *TNL section always contains the original input data provided on input. These data items are identified below in regular typeface. In addition, SNS may optionally return the TNRC return code shown in bold typeface.

```
*TNL{                              TN Line
    REC{                           Record {1-N}, where N = the number of TNL
                                   requests
        CTL{                       Control data
            CTC = B;               Control code: Build
            TNRC = ...;            TN return code {0-N}, where N = the number of
                                   TN processing conditions
        }                          End CTL aggregate
        ACL{                       Action line
            ACT = N;               Action code: New
            TN{                    Telephone number
                NPA = ...;         NPA
                NXX=...;     NXX
                LINE=...;    Line number
            }                End TN aggregate
        }                    End ACL aggregate
    }                        End REC aggregate
}%                           End *TNL section
```

On a response, SNS may include a TN Return Code if it attempted to process the order update. The TNRC tag supplied in the CTL aggregate indicates whether SNS was successful in processing the TN order update.

> **NOTE**
>
> As explained earlier, SNS does not perform partial order updates in the TNLIST database. As a result, the TNLIST database is *not* actually updated when an order error occurs, despite any "successful" return codes provided in the response message. The purpose of these return codes is to identify those order updates that would have been successful, had other errors not occurred.

See Appendix D for a description of the possible TNRC return codes.

5.3.2.4 Selected Inventory (*SELINV) Section

The presence of the *SELINV section in the response message indicates that SNS could not automatically return unused network resources that were selected during the negotiation dialogue. The *SELINV section identifies the remaining selected inventory in question. This section is returned after the ORDCMP contract has been invoked for the last order that was associated with the service request.

The data items provided in the *SELINV section are as follows:

```
*SELINV{                     Selected network resource inventory
    TNI{                     TN inventory selected
        TN{                  Telephone number {1-N}, where N=the number
                             of individual TNs selected
            NPA=...;         NPA
            NXX=...;         NXX
            LINE=...;        Line number
        }                     End TN aggregate
        TNRNG{                TN range {1-N}, where N = the number of groups
                              of consecutive TNs selected
            NPA = ...;        NPA
            NXX = ...;        NXX
            LOLN = ...;       Low line number
            HILN = ...;       High line number
        }                     End TNRNG aggregate
    }                         End TNI aggregate
}%                            End *SELINV section
```

5.3.2.5 Order Status Information (*ORDINFO) Section

The *ORDINFO section identifies orders in the negotiation dialogue that errored in SNS upon completion or cancellation. This section is returned after the ORDCMP contract has been invoked for the last order that was associated with the service request. The *ORDINFO section is supplied only when the *SELINV section is returned.

```
*ORDINFO{                     Order Status Information
    ORD{                      Order {1-N}, where N = the number of orders in
                              the negotiation dialogue that errored
        ORDID{                Order identifier
            ORDTYP = ...;     Order type
            ORDNUM = ...;     Order number
            CORSFX = ...;     Correction suffix [opt]
        }                     End ORDID aggregate
        ORDPASS = ...;        Order pass
                              < CAN = Cancelled,
                                PCN = Post completion >
    }                         End ORD aggregate
}%                            End *ORDINFO section
```

5.4 Hold Contract (HOLDSR)

The Hold contract holds the service request in SNS. Whenever the negotiation dialogue is put on hold, SNS refrains from returning any responses related to this negotiation until the service request is retrieved (via the RTRVSR contract).

This contract consists of a request and a response message.

- The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section [opt]
*HOLD	Hold Data section [opt]

The *HOLD section may contain any FCIF-conforming structure supplied by the contract invoker to save data about the negotiation.

- The FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section [opt]
*SNSHDR	SNS Header section
*HOLD	Hold data section [opt]

If the hold was successful, SNS does not return a *UMSG or *HOLD section. However, if the hold was not successful, SNS returns a *UMSG section, along with the *HOLD section optionally provided in the request message.

5.5 Retrieve Contract (RTRVSR)

The Retrieve contract serves two purposes:

1. Takes a held negotiation dialogue off hold so that SNS may resume returning responses for any outstanding requests related to this negotiation. In addition, SNS returns the negotiation data that was previously held when the Hold contract was last invoked. It also returns the status of each service order associated with the negotiation dialogue.
2. Obtains the service specification data for one or more issued service orders associated with the service request.

> **NOTE**
> This capability has limited use in SNS 2.0 because of the abridged Service Specification contract. In future SNS releases, the Service Specification contract will be expanded to include more service specification data.

This contract consists of a request and a response message.

- The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section [opt]
*RSRD	Request for Service Request Data

The *RSRD section indicates which retrieve options are requested.

- The FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section [opt]
*SNSHDR	SNS Header section
*SRDR	Service Request Data Response
*HOLD	Hold Data [opt]
*SVCSPEC	Service Specification section [opt]
*ORDINFO	Order Status Information section [opt]

The *SRDR section is always returned to summarize the results of processing the Retrieve contract. When SNS takes the negotiation off hold, it returns the *HOLD data supplied in the last Hold contract invocation. If service specification data was requested for issued orders, SNS supplies this data in the *SVCSPEC section. In either case, SNS also returns the *ORDINFO section to indicate the status of each order involved in the service request (or each requested order), except when orders were not issued.

5.5.1 Retrieve Contract Request Message

The request message contains the data needed to take the negotiation dialogue off hold and/or retrieve service specification data.

5.5.1.1 Request for Service Request Data (*RSRD) Section Format

This FCIF section identifies which retrieve options to perform. The data items for the *RSRD section are:

```
*RSRD{                      Request for Service Request Data
    CTL{                    Control data
        OFFHOLD=...;        Off hold option <Y=Yes> [opt]
        SVCSPEC=...;        Retrieve service specification data flag
                            <S=Retrieve data for specific order(s),
                              A=Retrieve data for all orders> [opt]
    }                       End CTL aggregate
    ACL{                    Action line {0-1}
        ORDID{              Order identifier {1-N}, where N=the number of
                            orders to retrieve when SVCSPEC=S
            ORDTYP=...;     Order type
            ORDNUM=...;     Order number
        }                   End ORDID aggregate
    }                       End ACL aggregate
}%                          End *RSRD section
```

The CTL aggregate identifies certain control information. It may contain one or both of the following options:

- The off hold option (OFFHOLD=Y) indicates that SNS should take the negotiation dialogue off hold.
- The retrieve service specification flag indicates that SNS should retrieve the service specification data for one or more orders (SVCSPEC=S) or all orders (SVCSPEC=A) associated with the negotiation dialogue.

When SVCSPEC = S, an ACL aggregate is required to identify the desired order(s).

5.5.2 Retrieve Contract Response Message

The response message contains the results of the request to retrieve service request data.

5.5.2.1 Service Request Data Response (*SRDR) Section Format

The data items for the *SRDR section are:

```
*SRDR{                        Service Request Administration Response
    CTL{                      Control data
        OFFHOLD=...;          Off hold option <Y=Yes> [opt]
        SVCSPEC=...;          Retrieve service specification data flag
                              <S=Retrieve data for specific order(s),
                                A=Retrieve data for all orders> [opt]
        IERC=...;             Input error return code {0-N}, where N=the
                              number of input errors detected
        RVRC=...;             Retrieve return code {0-2}
    }                         End CTL aggregate
    ACL{                      Action line {0-1}
        ORDID{                Order identifier {1-N}, where N=the number of
                              orders to retrieve when SVCSPEC=S
            ORDTYP=...;       Order type
            ORDNUM=...;       Order number
            CORSFX=...;       Correction suffix [opt]
        }                     End ORDID aggregate
    }                         End ACL aggregate
}%                            End *SRDR section
```

There are two return codes that may be included in the CTL aggregate data. In the general format above, they are shown in bold typeface under the CTL aggregate. The return codes are:

- *Input Error Return Code (IERC)*

  The contract invoker is responsible for providing SNS with a valid request message that conforms to the specifications outlined in this document. However, if SNS detects a critical input error condition that prevents it from continuing its processing, it notifies the contract invoker of the problem via the IERC tag. The value associated with the tag indicates the error condition. This tag may appear multiple times for multiple input errors.

- *Retrieve Return Code (RVRC)*

  The RVRC tag indicates whether SNS was successful in taking the negotiation dialogue off hold and in retrieving the requested service specification data. This tag may appear multiple times for multiple processing conditions.

See Appendix D for a description of the possible RVRC values.

5.5.2.2 Hold Data (*HOLD) Section

This FCIF section is identical to the *HOLD section described in the Hold contract request message.

5.5.2.3 Service Specification (*SVCSPEC) Section

This FCIF section is identical to the *SVCSPEC section described in the Service Specification Contract request message. It is returned whenever a request is made to retrieve the service specification data for issued orders.

5.5.2.4 Order Status Information (*ORDINFO) Section

This FCIF section indicates the current status (in SNS) of one or more orders involved in the negotiation dialogue. When the negotiation dialogue is taken off hold, the *ORDINFO section contains the status of each issued order associated with the negotiation. When only service specification data was requested, the *ORDINFO section contains only the status of those orders requested. This information is supplied to avoid corrections or cancellations on completed or canceled orders.

```
*ORDINFO{                Order Status Information
   ORD{                  Order {1-N}, where N = the number of orders
                         involved in the negotiation dialogue
      ORDID{             Order identifier
         ORDTYP = ...;   Order type
         ORDNUM = ...;   Order number
         CORSFX = ...;   Correction suffix [opt]
      }                  End ORDID aggregate
```

```
        ORDPASS=...;            Order pass
                                <CAN=Cancelled,
                                 PCN=Post-completion,
                                 PRE=Pre-completion>
    }                           End ORD aggregate
}%                              End *ORDINFO section
```

6. System Administration Table Maintenance Contracts

All system administration contracts are classified as *Table Maintenance.* Table Maintenance contracts can be one of two types, inquiry or update. The difference between inquiry contracts and update contracts is that only update contracts alter the contents of the database tables. An inquiry contract accesses the database tables and returns the data being requested to the contract invoker. An update contract provides the means to alter the contents of database tables through build, remove, and change functions. The inquiry contracts are:

- Inquiry of TNLIST data (INQTNL)
- Inquiry of Feature Group Catalog, Feature Catalog, and Name Description data (INQCAT)
- Inquiry of Network Element and Switch Parameter data or Switch Feature Exception data (INQNED)
- Inquiry of CPE data (INQCPE)
- Inquiry of Feature Package data (INQFPK)

The table maintenance update contracts are:

- Build, remove, or change data associated with a TNLIST Group (MTCTNL)
- Build, remove, or change Series Completion Hunt Groups and their associated telephone numbers (MTCHNT)
- Change Feature Group Catalog data, Feature Catalog data, and Name Description data (MTCCAT)
- Build, remove, or change data associated with Customer Premises Equipment (MTCCPE)
- Build, remove, or change Feature Packages and their associated data (MTCFPK)
- Build, remove, or change Network Element data (MTCNED)
- Build, remove, or change Switch Feature Exception data (MTCNED).

The following sections describe the System Administration Table Maintenance contracts. These descriptions are organized by type of data:

- TNLIST Maintenance contracts (INQTNL, MTCTNL, and MTCHNT)
- Catalog Data Maintenance contracts (INQCAT and MTCCAT)
- CPE Data Maintenance contracts (INQCPE and MTCCPE)
- Feature Package Data Maintenance contracts (INQFPK and MTCFPK)
- Switch-Related Data Maintenance contracts (INQNED and MTCNED).

6.1 TNLIST Maintenance Contracts

This section presents the specifications for the three TNLIST Maintenance contracts. These contracts support the build, remove, change, and inquiry functions for the Centrex Group, TN, and Hunting data residing in the TNLIST database.

The first contract, INQTNL, performs inquiries on TN data stored in a service type and/or hunt group. The second, MTCTNL, performs the build, remove, and change functions for the regular Centrex service type and its associated telephone numbers. Finally, the third contract, MTCHNT, performs the build, remove, and change functions for Series Completion Hunt Groups and their associated telephone numbers.

All three contracts may consist of *C1/*C2 headers, *PLHDR, *SNSHDR, *UMSG, and some combination of request/response sections. The following tables list the request and response sections used in the three contracts.

Table 6-1. INQTNL Contract Sections

Request Sections	Response Sections
*C1	*C2
*PLHDR	*PLHDR
*SNSHDR [opt]	*SNSHDR
	*UMSG [opt]
*RGRP	*GRPR
*RMTN	*MTNR
*RSCH	*SCHR

Table 6-2. MTCTNL Contract Sections

Request Sections	Response Sections
*C1	*C2
*PLHDR	*PLHDR
*SNSHDR [opt]	*SNSHDR
	*UMSG [opt]
*RGRP	*GRPR
*RMTN	*MTNR

Table 6-3. MTCHNT Contract Sections

Request Sections	Response Sections
*C1	*C2
*PLHDR	*PLHDR
*SNSHDR [opt]	*SNSHDR
	*UMSG [opt]
*RSCH	*SCHR

The *C1/*C2 headers, *PLHDR, *SNSHDR, and *UMSG sections are described in Section 3 of this document. The following sections describe the various request/response sections used in the TNLIST Maintenance contracts.

A description of the tag names and code sets used in the definition of the TNLIST Maintenance contracts is provided in Appendix B. Appendix D lists possible return codes, and Appendix E presents examples of TNLIST Maintenance contracts.

6.1.1 *RGRP Section

The *RGRP section is required in the request message for TNLIST processing of the Centrex Group Inquiry (INQTNL) and the Centrex Group Maintenance (MTCTNL) functions. The data items for the *RGRP section are listed below. This listing does not indicate whether the items are optional, conditional, or required. This information is supplied in subsequent sections.

```
*RGRP{
  REC{                          Record data {1-N}
     CTL{                       Control data {1}
        CTC=...;                Control code
        MODE=...;               Contract origin
        ALLSTAT=...;            Delete all TN entries
     }                          End CTL
     ACL{                       Action line {1-2}
        ACT=...;                Action code
        TIECODE=...;            Centrex customer ID
        GROUP{                  Centrex group data
           CLLI=...;            Network element CLLI code
           CTX=...;             Centrex group ID
           MAINTN{              Main TN data
              NPA=...;          TN area code
              NXX=...;          TN exchange code
              LINE=...;         TN line number
           }                    End MAINTN
           GRPTYP=...;          Group type
           INTCPT=...;          Intercept code
           LISTTHRS=...;        List threshold
           RCDATA{              Recent change data {1}
              PIC=...;          Primary interexchange carrier
              LCC=...;          Line class code
              CAT=...;          Customer access treatment
           }                    End RCDATA
        }                       End GROUP
     }                          End ACL
  }                             End REC
}%                              End *RGRP
```

The CTL aggregate contains control information. In particular, the control code (CTC) is used to determine the type of processing to be performed:

Inquiry for TNLIST Centrex Group data	(CTC = I)
Build TNLIST Centrex Group data	(CTC = B)
Remove TNLIST Centrex Group data	(CTC = R)
Change TNLIST Centrex Group data	(CTC = C)

The ACL aggregate contains the specific information required to perform the TNLIST Centrex Group inquiry/maintenance procedure. The information that SNS requires in each ACL aggregate varies, depending upon the CTC that appears in the CTL aggregate.

For a TNLIST Centrex Group inquiry/maintenance procedure, the action code (ACT) within the ACL aggregate may identify the type of ACL: old (O), new (N), or replace (R). The following types of ACLs are required, based on the CTC value:

- When CTC = I, an old ACL only is assumed.
- When CTC = B, a new ACL only is required.
- When CTC = R, an old ACL only is required.
- When CTC = C, both an old and new ACL are required, or a single replace ACL is required.

The following subsections explain how the *RGRP section should be constructed, depending on the type of control data provided in the CTL aggregate.

6.1.1.1 *RGRP Section Format: TNLIST Centrex Group Inquiry (INQTNL)

SNS uses the information provided in the *RGRP section of the INQTNL request message to determine the level of data returned by the inquiry. If only the TIECODE has been entered, then all of the groups associated with the specified customer will be returned. If the combination CLLI/CTX is entered, then only that data directly associated with the specified group is returned. Finally, if the MAINTN is entered, then only the data associated with that main telephone number is returned.

The following data items may appear in the *RGRP section of the INQTNL request message:

```
*RGRP{
  REC{                        Record data {1-N}
    CTL{                      Control data {1}
      CTC = I;                Control code
    }
    ACL{                      Action line {1}
      TIECODE = ...;          Centrex customer ID [opt]
      GROUP{                  Centrex group data [opt]
        CLLI = ...;           Network element CLLI code [conditional]
        CTX = ...;            Centrex group ID [conditional]
        MAINTN{               Main TN data [conditional]
          NPA = ...;          TN area code
          NXX = ...;          TN exchange code
          LINE = ...;         TN line number
        }                     End MAINTN
      }                       End GROUP
    }                         End ACL
  }                           End REC
}%                            End *RGRP
```

6.1.1.2 *RGRP Section Format: TNLIST Group Maintenance (MTCTNL)

SNS uses the information provided in the *RGRP section of the MTCTNL request message to build, remove, or change a specific TNLIST Centrex group. The format and actions depend on the control code (CTC) specified in the control data (CTL) aggregate.

6.1.1.2.1 *RGRP Section Format: TNLIST Group BUILD

To build a TNLIST group, the following *RGRP section information should be provided.

```
*RGRP{
  REC{                        Record data {1-N}
    CTL{                      Control data {1}
      CTC = B;                Control code
      MODE = ...;             Contract origin
    }                         End CTL
    ACL{                      Action line {1}
      ACT = N;                Action code
      TIECODE = ...;          Centrex customer ID [opt]
```

```
          GROUP{              Centrex group data {1}
            CLLI=...;         Network element CLLI code
            CTX=...;          Centrex group ID
            MAINTN{           Main TN data
              NPA=...;        TN area code
              NXX=...;        TN exchange code
              LINE=...;       TN line number
            }                 End MAINTN
            GRPTYP=...;       Group type
            INTCPT=...;       Intercept code [opt]
            LISTTHRS...;      List threshold [opt]
            RCDATA{           Recent change data [opt] {1}
              PIC=...;        Primary interexchange carrier
              LCC=...;        Line class code
              CAT=...;        Customer access treatment
            }                 End RCDATA
          }                   End GROUP
        }                     End ACL
  }                           End REC
}%                            End *RGRP
```

6.1.1.2.2 *RGRP Section Format: TNLIST Group REMOVE

A TNLIST Centrex group can be removed in different ways. If just the TIECODE is entered, then an attempt is made to remove all groups associated with the TIECODE. If just the CLLI/CTX or MAINTN is entered, then an attempt to remove only that specified group is made. If the ALLSTAT tag is set to "Y," the delete is performed regardless of the TN status.

```
*RGRP{
  REC{                        Record data {1-N}
    CTL{                      Control data {1}
      CTC=R;                  Control code
      ALLSTAT=...;            Delete all TN entries [opt]
    }                         End CTL
    ACL{                      Action line {1}
      ACT=O;                  Action code
      TIECODE=...;            Centrex customer ID [opt]
      GROUP{                  Centrex group data [opt]
```

```
        CLLI=...;         Network element CLLI code [conditional]
        CTX=...;          Centrex group ID [conditional]
        MAINTN{           Main TN data [conditional]
          NPA=...;        TN area code
          NXX=...;        TN exchange code
          LINE=...;       TN line number
        }                 End MAINTN
      }                   End GROUP
    }                     End ACL
  }                       End REC
}%                        End *RGRP
```

6.1.1.2.3 *RGRP Section Format: TNLIST Group CHANGE

To change a TNLIST group, the existing key information — as defined in the *Service Negotiation Support (SNS) Database Administration Guide* (BR 007-560-406) — must be provided in either the "O" ACL or "R" ACL aggregate. The key information for the *RGRP section is defined as the valid combinations of TIECODE, CLLI, CTX, and MAINTN, which are used to identify specific TNLIST Centrex groups.

If the "O" ACL aggregate is used, then new entries must be supplied in the "N" ACL aggregate. The "O" ACL aggregate must match the values in the database or no changes will be made. With this method of change, the key information may be modified.

```
*RGRP{
  REC{                    Record data {1-N}
    CTL{                  Control data {1}
      CTC=C;              Control code
    }                     End CTL
    ACL{                  Action line {1}
      ACT=O;              Action code
      TIECODE=...;        Centrex customer ID [opt]
      GROUP{              Centrex group data [conditional]
        CLLI=...;         Network element CLLI code
        CTX=...;          Centrex group ID
        MAINTN{           Main TN data
          NPA=...;        TN area code
          NXX=...;        TN exchange code
```

```
         LINE=...;  TN line number
       }            End MAINTN
       GRPTYP=...;  Group type [opt]
       INTCPT=...;  Intercept code [conditional]
       LISTTHRS...; List threshold [opt]
       RCDATA{      Recent change data [conditional] {1}
         PIC=...;   Primary interexchange carrier
         LCC=...;   Line class code
         CAT=...;   Customer access treatment
       }            End RCDATA
     }              End GROUP
   }                End ACL
   ACL{             Action line {1}
     ACT=N;         Action code
     TIECODE=...;   Centrex customer ID [opt]
     GROUP{         Centrex group data {1}
       CLLI=...;    Network element CLLI code
       CTX=...;     Centrex group ID
       MAINTN{      Main TN data
         NPA=...;   TN area code
         NXX=...;   TN exchange code
         LINE=...;  TN line number
       }            End MAINTN
       GRPTYP=...;  Group type [opt]
       INTCPT=...;  Intercept code [opt]
       LISTTHRS...; List threshold [opt]
       RCDATA{      Recent change data [opt] {1}
         PIC=...;   Primary interexchange carrier
         LCC=...;   Line class code
         CAT=...;   Customer access treatment
       }            End RCDATA
     }              End GROUP
   }                End ACL
 }                  End REC
}%                  End *RGRP
```

If the "R" ACL aggregate is used, then the specified tags are used to update the database. This type of change is unconditional and requires that the key information only match the values in the database. With this method of change, the key information may *not* be modified.

```
*RGRP{
  REC{                          Record data
     ACL{                       Action line {1}
        ACT=R;                  Action code
        TIECODE=...;            Centrex customer ID [opt]
        GROUP{                  Centrex group data [conditional]
           CLLI=...;            Network element CLLI code
           CTX=...;             Centrex group ID
           MAINTN{              Main TN data
              NPA=...;          TN area code
              NXX=...;          TN exchange code
              LINE=...;         TN line number
           }                    End MAINTN
           GRPTYP=...;          Group type [opt]
           INTCPT=...;          Intercept code [opt]
           LISTTHRS...;         List threshold [opt]
           RCDATA{              Recent change data [opt] {1}
              PIC=...;          Primary interexchange carrier
              LCC=...;          Line class code
              CAT=...;          Customer access treatment
           }                    End RCDATA
        }                       End GROUP
     }                          End ACL
  }                             End REC
}%                              End *RGRP
```

6.1.2 *GRPR Section

The *GRPR section returned in the response message presents the results of the INQTNL and MTCTNL contracts. It may contain customer, group, and/or feature information. It may also identify errors and discrepancies.

6.1.2.1 *GRPR Section Format: TNLIST Centrex Group Response (INQTNL/MTCTNL)

The data items for the *GRPR section are shown in the following general format. The items shown in regular type reflect the data provided in the *RGRP section of the original request. The original input is always provided in the corresponding ACL aggregate of the *GRPR section of the response contract. The items shown in bold type represent the additional response data. The first group of items appear in the CTL aggregate and contain the return codes for the attempted action. The second group appears in the GRPDATA aggregate, which contains any additional data returned from the TNLIST database tables.

```
*GRPR{
  REC{                          Record data {1-N}
     CTL{                       Control data {1}
        CTC=...;                Control code
        MODE=...;               Contract origin
        ALLSTAT=...;            Delete all TN entries
        IERC=...;               Input error return code {1-N}
        GRPRC=...;              Centrex database validation return code {1}
     }                          End CTL
     ACL{                       Action line {1-2}
        ACT=...;                Action code
        TIECODE=...;            Centrex customer ID
        GROUP{                  Centrex group data {1-N}
           CLLI=...;            Network element CLLI code
           CTX=...;             Centrex group ID
           MAINTN{              Main TN data
              NPA=...;          TN area code
              NXX=...;          TN exchange code
              LINE=...;         TN line number
           }                    End MAINTN
           GRPTYP=...;          Group type
           INTCPT=...;          Intercept code
           LISTTHRS=...;        List threshold
           RCDATA{              Recent change data [opt] {1}
              PIC=...;          Primary interexchange carrier
              LCC=...;          Line class code
              CAT=...;          Customer access treatment
           }                    End RCDATA
        }                       End GROUP
     }                          End ACL
     GRPDATA{                   Group data {1}
        TIECODE=...;            Centrex customer ID
        GROUP{                  Centrex group data {1-N}
           CLLI=...;            Network element CLLI code
           CTX=...;             Centrex group ID
           MAINTN{              Main TN data
              NPA=...;          TN area code
              NXX=...;          TN exchange code
              LINE=...;         TN line number
```

```
            }                    End MAINTN
            GRPTYP=...;          Group type
            INTCPT=...;          Intercept code
            LISTTHRS=...;        List threshold
            RCDATA{              Recent change data [opt] {1}
                PIC=...;         Primary interexchange carrier
                LCC=...;         Line class code
                CAT=...;         Customer access treatment
            }                    End RCDATA
        }                        End GROUP
    }                            End GRPDATA
  }                              End REC
}%                               End *GRPR
```

6.1.2.2 CTL Aggregate Return Codes

There are two return codes that may be included with the CTL aggregate data. In the general format above they are shown in bold face type. These return codes are:

- *Input Error Return Code (IERC)*

  The contract invoker is responsible for providing SNS with a valid TNLIST Centrex group request message that conforms to the specifications outlined in this document. However, if SNS detects an input error condition that prevents it from continuing its processing, it notifies the contract invoker via the IERC tag. The value associated with the tag indicates the error condition. The tag may appear multiple times to identify multiple input errors.

- *Section Return Code (GRPRC)*

  The GRPRC tag indicates the results of the attempted database processing. It informs the contract invoker of the conditions found in the TNLIST database and whether or not the requested action has taken place. The tag may appear only once.

If the request message was an Inquiry, then the appropriate IERC or GRPRC codes would indicate a failure to retrieve the requested data items.

If the request message was a Build, a failure would be reported by the appropriate IERC or GRPRC codes. A success would result in the addition of the data to the appropriate TNLIST database tables.

If the request message was a Remove, a failure would be reported by the appropriate IERC or GRPRC codes. A success would result in the removal of the data from the appropriate TNLIST database tables.

Finally, if the request message was a Change, a failure would be reported by the appropriate IERC or GRPRC codes. A success would result in the appropriate modifications to the TNLIST database tables.

6.1.3 *RMTN Section

The *RMTN section is required in the request message for TNLIST processing of the TN Inquiry (INQTNL) and the TN Maintenance (MTCTNL) functions. The items for the *RMTN section are listed below. This listing does not indicate whether the items are optional, conditional, or required. This information is supplied in subsequent sections.

```
*RMTN{
  REC{                      Record data {1-N}
     CTL{                   Control data {1}
        CTC=...;            Control code
        MODE=...;           Contract origin
        ALLSTAT=...;        Delete all TN entries
     }                      End CTL
     ACL{                   Action line {1-2}
        ACT=...;            Action code
        GROUP{              Centrex group data {1}
           CLLI=...;        Network element CLLI code
           CTX=...;         Centrex group ID
           MAINTN{          Main TN data
              NPA=...;      TN area code
              NXX=...;      TN exchange code
              LINE=...;     TN line number
           }                End MAINTN
        }                   End GROUP
        TNLIST{             TN list data {1-N}
           NPA=...;         TN area code
           NXX=...;         TN exchange code
           LOLN=...;        Low line number of TN range
           HILN=...;        High line number of TN range
           STAT=...;        Current TN status
           TYPE=...;        Type of service for specific TN
           SUBTYP=...;      TN attributes
        }                   End TNLIST
```

```
      }                          End ACL
   }                             End REC
}%                               End *RMTN
```

The CTL aggregate contains control information. In particular, the control code (CTC) is used to determine the type of processing to be performed.

Inquiry for TNLIST TN data	(CTC = I)
Build TNLIST TN data	(CTC = B)
Remove TNLIST TN data	(CTC = R)
Change TNLIST TN data	(CTC = C)

The ACL aggregate contains the specific information required to perform the TNLIST TN inquiry/maintenance procedure. The information that SNS requires in each ACL aggregate varies, depending on the CTC value that appears in the CTL aggregate.

For a TNLIST TN inquiry/maintenance procedure, the action code (ACT) within the ACL aggregate may identify the type of ACL: old (O), new (N), or replace (R). The following types of ACLs are required, based on the CTC value:

- When CTC = I, an old ACL only is assumed.
- When CTC = B, an new ACL only is required.
- When CTC = R, an old ACL only is required.
- When CTC = C, both an old and new ACL are required, or a single replace ACL is required.

The following subsections explain how the *RMTN section should be constructed, depending on the type of control data provided in the CTL aggregate.

6.1.3.1 *RMTN Section Format: TNLIST TN Inquiry (INQTNL)

SNS uses the information provided in the *RMTN section of the INQTNL request message to determine the level of data returned by the inquiry. If the group data only has been entered, then the TN range information is returned for all ranges of TNs associated with the specified group. If the NPA, NXX, LOLN, and HILN are entered in addition to the group data, then all of the individual telephone number information is returned for the specified range.

The following data items may appear in the *RMTN section of the INQTNL request message:

```
*RMTN{
  REC{                        Record data {1-N}
     CTL{                     Control data {1}
        CTC=I;                Control code
     }                        End CTL
     ACL{                     Action line {1}
        GROUP{                Group data [conditional] {1}
           CLLI=...;          Network element CLLI code
           CTX=...;           Centrex group ID
           MAINTN{            Main TN data
              NPA=...;        TN area code
              NXX=...;        TN exchange code
              LINE=...;       TN line number
           }                  End MAINTN
        }                     End GROUP
        TNLIST{               TN list [opt] {1-N}
           NPA=...;           TN area code
           NXX=...;           TN exchange code
           LOLN=...;          Low line number of TN range
           HILN=...;          High line number of TN range
        }                     End TNLIST
     }                        End ACL
  }                           End REC
}%                            End *RMTN
```

6.1.3.2 *RMTN Section Format: TNLIST TN Maintenance (MTCTNL)

SNS uses the information provided in the *RMTN section of the MTCTNL request message to build, remove, or change a specific TNLIST TN or a range of TNs. The formats and actions depend on the control code (CTC) value specified in the control data (CTL) aggregate.

6.1.3.2.1 *RMTN Section Format: TNLIST TN Build

To build a TNLIST TN or range of TNs, the following section information should be provided:

```
*RMTN{
  REC{                        Record data {1-N}
     CTL{                     Control data {1}
        CTC=B;                Control code
```

```
         MODE=...;              Contract origin
      }                         End of CTL
      ACL{                      Action line {1}
         ACT=N;                 Action code
         GROUP{                 Centrex group data [conditional] {1}
            CLLI=...;           Network element CLLI code
            CTX=...;            Centrex group ID
            MAINTN{             Main TN data
               NPA=...;         TN area code
               NXX=...;         TN exchange code
               LINE=...;        TN line number
            }                   End of MAINTN
         }                      End of GROUP
         TNLIST{                TN list {1-N}
            NPA=...;            TN area code
            NXX=...;            TN exchange code
            LOLN=...;           Low line number of TN range
            HILN=...;           High line number of TN range [opt]
            STAT=...;           Current TN status
            TYPE=...;           Type of service for specific TN
            SUBTYPE=...;        TN attributes [opt]
         }                      End of TNLIST
      }                         End of ACL
   }                            End of REC
}%                              End of *RMTN
```

6.1.3.2.2 *RMTN Section Format: TNLIST TN Remove

When attempting to remove a TNLIST TN or range of TNs, there are two possibilities. If just the group data is entered, an attempt to remove every TN associated with that group is made. If the NPA, NXX, LOLN, and HILN are entered as well, then an attempt to remove only those TNs associated with the specified range is made. If the ALLSTAT tag is set to "Y," the delete is performed regardless of TN status.

```
*RMTN{
   REC{                         Record data {1-N}
      CTL{                      Control data {1}
         CTC=R;                 Control code
         ALLSTAT=...;           Delete all TN entries [opt]
      }                       End of CTL
      ACL{                    Action line {1}
         ACT = O;             Action code
         GROUP{               Group data [conditional] {1}
            CLLI = ...;       Network element CLLI code
            CTX = ...;        Centrex group ID
            MAINTN{           Main TN data
               NPA = ...;     TN area code
               NXX = ...;     TN exchange code
               LINE = ...;    TN line number
            }                 End of MAINTN
         }                    End of GROUP
         TNLIST{              TN list [opt] {1-N}
            NPA = ...;        TN area code
            NXX = ...;        TN exchange code
            LOLN = ...;       Low line number of TN range
            HILN = ...;       High line number of TN range
         }                    End of TNLIST
      }                       End of ACL
   }                          End of REC
}%                            End of *RMTN
```

6.1.3.2.3 *RMTN Section Format: TNLIST TN Change

When attempting to change a TNLIST TN or range of TNs, the existing key information must be provided in either the "O" ACL or "R" ACL aggregate. The key information for the *RMTN section – as defined in the *Service Negotiation Support (SNS) Database Administration Guide* (BR 007-560-406) – is any valid combination of CLLI, CTX, and MAINTN that identifies the group and the NPA, NXX, LOLN, and HILN that optionally identify a specific set of telephone numbers.

If the "O" ACL aggregate is used, then new entries must be supplied in the "N" ACL aggregate. The "O" ACL aggregate must match the values in the database, or no changes will be made. With this method of change, the key information may be modified.

```
*RMTN{
  REC{                        Record data {1-N}
    CTL{                      Control data
      CTC=C;                  Control code
    }                         End CTL
    ACL{                      Action line {1}
      ACT=O;                  Action code
      GROUP{                  Centrex group data [conditional] {1}
        CLLI=...;             Network element CLLI code
        CTX=...;              Centrex group ID
        MAINTN{               Main TN data
          NPA=...;            TN area code
          NXX=...;            TN exchange code
          LINE=...;           TN line number
        }                     End MAINTN
      }                       End GROUP
      TNLIST{                 TN list data {1-N}
        NPA=...;              TN area code
        NXX=...;              TN exchange code
        LOLN=...;             Low line number of TN range
        HILN=...;             High line number of TN range
        STAT=...;             Current TN status
        TYPE=...;             Type of service for specific TN
        SUBTYP=...;           TN attributes
      }                       End TNLIST
    }                         End ACL
    ACL{                      Action line {1}
      ACT=N;                  Action code
      GROUP{                  Centrex group data [conditional] {1}
        CLLI=...;             Network element CLLI code
        CTX=...;              Centrex group ID
        MAINTN{               Main TN data
          NPA=...;            TN area code
          NXX=...;            TN exchange code
          LINE=...;           TN line number
        }                     End MAINTN
      }                       End GROUP
      TNLIST{                 TN list data {1-N}
        NPA=...;              TN area code
```

```
            NXX=...;          TN exchange code
            LOLN=...;         Low line number of TN range
            HILN=...;         High line number of TN range
            STAT=...;         Current TN status
            TYPE=...;         Type of service for specific TN
            SUBTYP=...;       TN attributes
         }                    End TNLIST
      }                       End ACL
   }                          End REC
}%                            End *RMTN
```

If the "R" ACL aggregate is used, then the specified tags are used to update the database. This type of change is unconditional and requires that the key information only match the database. With this method of change, the key information may *not* be modified.

```
*RMTN{
   REC{                       Record data
      ACL{                    Action line {1}
         ACT=R;               Action code
         GROUP{               Centrex group data [conditional] {1}
            CLLI=...;         Network element CLLI code
            CTX=...;          Centrex group ID
            MAINTN{           Main TN data
               NPA=...;       TN area code
               NXX=...;       TN exchange code
               LINE=...;      TN line number
            }                 End MAINTN
         }                    End GROUP
         TNLIST{              TN list data {1-N}
            NPA=...;          TN area code [opt]
            NXX=...;          TN exchange code [opt]
            LOLN=...;         Low line number of TN range [opt]
            HILN=...;         High line number of TN range [opt]
            STAT=...;         Current TN status [opt]
            TYPE=...;         Type of service for specific TN [opt]
            SUBTYP=...;       TN attributes [opt]
         }                    End TNLIST
      }                       End ACL
```

```
  }                          End REC
}%                           End *RMTN
```

6.1.4 *MTNR Section

The *MTNR section returned in the response message presents the results of the INQTNL and MTCTNL contracts. It may contain information on TNs or ranges of TNs. It may also identify errors and discrepancies.

6.1.4.1 *MTNR Section Format: TNLIST TN Response (INQNTL/MTCTNL)

The data items for the *MTNR section are shown in the following general format. The items shown in regular type reflect the data provided in the *RMTN section of the original request. The original input is always provided in the corresponding ACL aggregate of the *MTNR section of the response contract. The items shown in bold type represent the response data. The first group of items appears in the CTL aggregate and contain the return codes for the attempted action. The second group appears in the TNDATA aggregate and contains any additional data returned from the TNLIST database tables. The third group appears in the TNLERR aggregate, which associates a specific TN or range of TNs with an error condition.

```
*MTNR{
  REC{                       Record data {1-N}
    CTL{                     Control data {1}
      CTC=...;               Control code
      MODE=...;              Contract origin
      ALLSTAT=...;           Delete all TN entries
      IERC=...;              Input error return code {1-N}
      MTNRC=...;             TNLIST database validation return code {1}
    }                        End CTL
    ACL{                     Action line {1-2}
      ACT=...;               Action code
      GROUP{                 Centrex group data {1}
        CLLI=...;            Network element CLLI code
        CTX=...;             Centrex group ID
        MAINTN{              Main TN data
          NPA=...;           TN area code
          NXX=...;           TN exchange code
          LINE=...;          TN line number
```

```
            }                       End MAINTN
        }                           End GROUP
        TNLIST{                     TN list data {1-N}
            NPA=...;                TN area code
            NXX=...;                TN exchange code
            LOLN=...;               Low line number of TN range
            HILN=...;               High line number of TN range
            STAT=...;               Current TN status
            TYPE=...;               Type of service for specific TN
            SUBTYP=...;             TN attributes
        }                           End TNLIST
    }                               End ACL
    TNDATA{                         TN data {1-N}
        NPA=...;                    TN area code
        NXX=...;                    TN exchange code
        LOLN=...;                   Low line number of TN range
        HILN=...;                   High line number of TN range
        STAT=...;                   Current TN status
        TYPE=...;                   Type of service for specific TN
        SUBTYP=...;                 TN attributes
    }                               End TNDATA
    TNLERR{                         TNLIST error data [opt] {1-N}
        NPA=...;                    TN area code
        NXX=...;                    TN exchange code
        LOLN=...;                   Low line number of TN range
        HILN=...;                   High line number of TN range
        TNERRCDE=...;               TNLIST aggregate error code
    }                               End TNLERR
  }                                 End REC
}%                                  End *MTNR
```

6.1.4.2 CTL Aggregate Return Codes

There are two return codes that may be included with the CTL aggregate data. In the general format above they are shown in bold type. These return codes are:

- *Input Error Return Code (IERC)*

  The contract invoker is responsible for providing SNS with a valid TNLIST TN request message that conforms to the specifications outlined in this document. However, if SNS detects an input error condition that prevents it from continuing its processing, it notifies the contract invoker via the IERC tag. The value associated with the tag indicates the error condition. The tag may appear multiple times to identify multiple input errors.

- *Section Return Code (MTNRC)*

  The MTNRC tag indicates the results of the attempted database processing. It informs the contract invoker of the conditions found in the TNLIST database and whether or not the requested action has taken place. The tag may appear only once.

If the request message was an Inquiry, then the appropriate IERC or MTNRC codes would indicate a failure to retrieve the requested data items.

If the request message was a Build, a failure would be reported by the appropriate IERC or MTNRC codes. A success would result in the addition of the data to the appropriate TNLIST database tables.

If the request message was a Remove, a failure would be reported by the appropriate IERC or MTNRC codes. A success would result in the removal of the data from the appropriate TNLIST database tables.

Finally, if the request message was a Change, a failure would be reported by the appropriate IERC or MTNRC codes. A success would result in the appropriate modifications to the TNLIST database tables.

6.1.4.3 TNLERR Aggregate Return Code

There is one return code that may be returned with the TNLERR aggregate. In the general format above it is shown in bold type along with the other elements of the TNLERR aggregate.

6.1.4.4 TNERRCDE Return Code

The TNERRCDE return code is returned for each individual TN that has an error associated with it.

6.1.5 *RSCH Section

The *RSCH section is required in the request message for TNLIST processing of Series Completion Hunt (SCH) Group Inquiry (INQTNL) and SCH Group Maintenance (MTCHNT) functions. The data items for the *RSCH section are listed below. This listing does not indicate whether the items are optional, conditional, or required. This information is supplied in subsequent sections.

```
*RSCH{
  REC{                     Record data {1-N}
     CTL{                  Control data {1}
        CTC=...;           Control code
        MODE=...;          Contract origin
     }                     End CTL
     ACL{                  Action line {1-2}
        ACT=...;           Action code
        GROUP{             Hunt group data {1}
           HTYP=...;       Hunting type
           TN{             TN data {1-N}
              NPA=...;     TN area code
              NXX=...;     TN exchange code
              LINE=...;    TN line number
           }               End TN
        }                  End GROUP
     }                     End ACL
  }                        End REC
}%                         End *RSCH
```

The CTL aggregate contains control information. In particular, the control code (CTC) is used to determine the type of processing to be performed:

Inquiry for TNLIST SCH data	(CTC = I)
Build TNLIST SCH data	(CTC = B)
Remove TNLIST SCH data	(CTC = R)
Change TNLIST SCH data	(CTC = C)

The ACL aggregate contains the specific information required to perform the TNLIST SCH inquiry/maintenance procedure. The information that SNS requires in each ACL aggregate varies, depending on the CTC value that appears in the CTL aggregate.

For a TNLIST SCH inquiry/maintenance procedure, the action code (ACT) within the ACL aggregate may identify the type of ACL: old (O), new (N), or replace (R). The following types of ACLs are required, based on the CTC value:

- When CTC = I, an old ACL only is assumed.
- When CTC = B, a new ACL only is required.
- When CTC = R, an old ACL only is required.

- When CTC = C, both an old and new ACL are required, or a single replace ACL is required.

The following subsections explain how the *RSCH section should be constructed, depending on the type of control data provided in the CTL aggregate.

6.1.5.1 *RSCH Section Format: TNLIST SCH Group Inquiry (INQTNL)

SNS uses the information provided in the *RSCH section of the INQTNL request message to determine which SCH group should be returned by the inquiry. The only inquiry available for SCH groups is via TN. It returns the entire hunt group.

The following data items may appear in the *RSCH section of the INQTNL request message:

```
*RSCH{
  REC{                          Record data {1-N}
     CTL{                       Control data {1}
        CTC=I;                  Control code
     }                          End CTL
     ACL{                       Action line {1}
        GROUP{                  Hunt group data {1}
           TN{                  TN data {1-N}
              NPA=...;          TN area code
              NXX=...;          TN exchange code
              LINE=...;         TN line number
           }                    End TN
        }                       End GROUP
     }                          End ACL
  }                             End REC
}%                              End *RSCH
```

6.1.5.2 *RSCH Section Format: TNLIST SCH Group Maintenance (MTCHNT)

SNS uses the information provided in the *RSCH section of the MTCHNT request message to build, remove, or change a specific TNLIST SCH group. The formats and actions depend on the control code (CTC) value specified in the control data (CTL) aggregate.

6.1.5.2.1 *RSCH Section Format: TNLIST SCH Group Build

To build an TNLIST SCH group the following section information should be provided.

```
*RSCH{
  REC{                        Record data {1-N}
     CTL{                     Control data {1}
        CTC=B;                Control code
        MODE=...;             Contract origin
     }                        End CTL
     ACL{                     Action line {1}
        ACT=N;                Action code
        GROUP{                Hunt group data {1}
           HTYP=...;          Hunting type
           TN{                TN data {1-N}
              NPA=...;        TN area code
              NXX=...;        TN exchange code
              LINE=...;       TN line number
           }                  End TN
        }                     End GROUP
     }                        End ACL
  }                           End REC
}%                            End *RSCH
```

6.1.5.2.2 *RSCH Section Format: TNLIST SCH Group Remove

When attempting to remove a TNLIST SCH group only one possibility exists. If the CTC =R, then the only required tags in the old ACL aggregate are the ACT tag and the TN aggregate for the PILOT_TN. This request removes the entire hunt group and all of its related data.

```
*RSCH{
  REC{                        Record data {1-N}
     CTL{                     Control data {1}
     CTC=R;                   Control code
     }                        End CTL
     ACL{                     Action line {1}
        ACT=O;                Action code
        GROUP{                Hunt group data
           TN{                TN data {1-N}
              NPA=...;        TN area code
              NXX=...;        TN exchange code
              LINE=...;       TN line number
```

```
            }                       End TN
          }                         End GROUP
        }                           End ACL
  }                                 End REC
}%                                  End *RSCH
```

6.1.5.2.3 *RSCH Section Format: TNLIST SCH Group Change

When attempting to change a TNLIST SCH group, the existing key information must be provided in either the "O" ACL or "R" ACL aggregate. The key information for the *RSCH section – as defined in the *Service Negotiation Support (SNS) Database Administration Guide* (BR 007-560-406) – is the PILOT_TN, which is the first TN of the Series Completion Hunt Group.

If the "O" ACL aggregate is used, then new entries must be supplied in the "N" ACL aggregate. The "O" ACL aggregate must match the values in the database or no changes will be made. With this method, the key information may be changed.

```
*RSCH{
  REC{                        Record data {1-N}
    CTL{                      Control data {1}
      CTC=C;                  Control code
    }                         End CTL
    ACL{                      Action line {1}
      ACT=O;                  Action code
      GROUP{                  Hunt group data
        HTYP=...;             Hunting type
        TN{                   TN data {1-N}
          NPA=...;            TN area code
          NXX=...;            TN exchange code
          LINE=...;           TN line number
        }                     End TN
      }                       End GROUP
    }                         End ACL
    ACL{                      Action line {1}
      ACT=N;                  Action code
      GROUP{                  Hunt group data
        HTYP=...;             Hunting type
        TN{                   TN data {1-N}
          NPA=...;            TN area code
          NXX=...;            TN exchange code
          LINE=...;           TN line number
        }                     End TN
      }                       End GROUP
    }                         End ACL
  }                           End REC
}%                            End *RSCH
```

If the "R" ACL aggregate is used, then the specified tags are used to update the database. This type of change is unconditional and requires that the key data only match the database. With this method of change, the key data may *not* be changed.

```
*RSCH{
  REC{                        Record data
    ACL{                      Action line {1}
      ACT=R;                  Action code
      GROUP{                  Hunt group data
        HTYP=...;             Hunting type [opt]
        TN{                   TN data {1-N}
          NPA=...;            TN area code
          NXX=...;            TN exchange code
          LINE=...;           TN line number
        }                     End TN
      }                       End GROUP
    }                         End ACL
  }                           End REC
}%                            End *RSCH
```

NOTE

The first occurrence of the TN aggregate in the "R" ACL aggregate must be the unchanged PILOT_TN of the TNLIST SCH group.

6.1.6 *SCHR Section

The *SCHR section returned in the response message presents the results of the INQTNL and MTCHNT contracts. It may contain TNLIST SCH group information. It may also identify errors and discrepancies.

6.1.6.1 *SCHR Section Format: TNLIST SCH Group Response (INQTNL/MTCHNT)

The data items for the *SCHR section are shown in the following general format. The items shown in regular type reflect the data provided in the *RSCH section of the original request. The original input is always provided in the corresponding ACL aggregate of the *SCHR section of the response contract. The items shown in bold type represent the response data. The first group of items appear in the CTL aggregate and contain the return codes for the attempted action. The second group appears in the HTNGRP aggregate and contain any additional data returned from the TNLIST database tables, as well as any hunting error conditions associated with the individual TNs.

```
*SCHR{
  REC{                        Record data {1-N}
    CTL{                      Control data {1}
      CTC=...;                Control code
```

```
        MODE=...;               Contract origin
        IERC=...;               Input error return code {1-N}
        SCHRC=...;              Hunt group database validation return code {1}
     }                          End CTL
     ACL{                       Action line {1}
        ACT=...;                Action code
        GROUP{                  Hunt group data {1}
           HTYP=...;            Hunting type
           TN{                  TN data {1-N}
              NPA=...;          TN area code
              NXX=...;          TN exchange code
              LINE=...;         TN line number
           }                    End TN
        }                       End GROUP
     }                          End ACL
     HTNGRP{                    Hunt group return data {1-N}
        CLLI=...;               Network element CLLI code
        HTYP=...;               Hunting type
        HTN{                    Hunt TN data {1-N}
           NPA=...;             TN area code
           NXX=...;             TN exchange code
           LINE=...;            TN line number
           HTNERR=...;          HTN aggregate error code
        }                       End HTN
     }                          End HTNGRP
  }                             End REC
}%                              End *SCHR
```

6.1.6.2 CTL Aggregate Return Codes

There are two return codes that may be included with the CTL aggregate data. In the general format above they are shown in bold type under the CTL aggregate. These return codes are:

- Input Error Return Code (IERC)

  The contract invoker is responsible for providing SNS with a valid TNLIST SCH group request message that conforms to the specifications outlined in this document. However, if SNS detects an input error condition that prevents it from continuing its processing, it notifies the contract invoker via the IERC tag. The value associated with the tag indicates the error condition. The tag may appear multiple times to identify multiple input errors.

- Section Return Code (SCHRC)

  The SCHRC tag indicates the results of the attempted database processing. It informs the contract invoker of the conditions found in the TNLIST database and whether or not the requested action has taken place. The tag may appear only once.

If the request message was an Inquiry, then the appropriate IERC or SCHRC codes would indicate a failure to retrieve the requested data items.

If the request message was a Build, a failure would be reported by the appropriate IERC or SCHRC codes. A success would result in the addition of the data to the appropriate TNLIST database tables.

If the request message was a Remove, a failure would be reported by the appropriate IERC or SCHRC codes. A success would result in the removal of the data from the appropriate TNLIST database tables.

Finally if the request message was a Change, a failure would be reported by the appropriate IERC or SCHRC codes. A success would result in the appropriate modifications to the TNLIST database tables.

6.1.6.3 HTN Aggregate Return Code

There is one return code that may be returned with the HTN aggregate. In the general format above it is shown in bold type along with the other elements of the HTN aggregate.

6.1.6.4 HTNERR Return Code

This return code is returned for each individual TN that has an error associated with it.

6.2 Catalog Data Maintenance Contracts

This section describes two contracts, INQCAT and MTCCAT, which are used to maintain catalog data. Note that catalog data encompasses feature group catalog, feature catalog, and feature name data. The INQCAT contract supports the inquiry of catalog data, while the MTCCAT contract supports the maintenance (change only) of catalog data. The change functionality of nonkey data only is supported, since the data for these tables is provided by Bellcore.

Both contracts consist of a request and a response message and share the *RCAT, *CATR, *RDES, and *DESR sections listed below.

- The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section (optional)
*RCAT	Request for Catalog Information section
*RDES	Request for Name Description Information section

- The FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section (optional)
*SNSHDR	SNS Header section
*CATR	Catalog Information Response section
*DESR	Name Description Information Response section

See Section 3 in this document for more information on the *C1, *C2, *PLHDR, *UMSG, and *SNSHDR FCIF sections.

6.2.1 General Contract Request Information

The contracts are designed so that multiple requests may be initiated in one contract invocation (request message). This is denoted by the REC aggregate that spans from {1-N}. Therefore, the INQCAT contract may contain multiple inquiry requests, while the MTCCAT contract may contain multiple requests for changes. These requests are structured in terms of the following FCIF sections.

The *RCAT section supports the maintenance of Feature Group Catalog and Feature Catalog Data:

- Inquiry of Feature Group Catalog, Feature Catalog, and Name Description Data (INQCAT)

- Change Feature Group Catalog and Feature Catalog Data (MTCCAT).

The *RDES section supports the maintenance of Feature Name Data:

- Inquiry of Name Description Data (INQCAT)
- Change Name Description Data (MTCCAT).

6.2.2 Request Catalog Data (*RCAT) Section

The *RCAT section contains the data needed to perform inquiries and changes to feature group catalog and feature catalog data.

6.2.2.1 General *RCAT Section Format

The data items† for the *RCAT section are:

```
*RCAT{                          Request for Catalog Data
    REC{                        Record {1-N}, where N equals the number of
                                Catalog Data Requests
        CTL{                    Control data
            CATOPT=...;         Catalog option:
                                <F=Feature Catalog Information,
                                  G=Feature Group Catalog Information>
            PROP=...;           Propagate option:
                                <Y=Yes,
                                  N=No,
                                  A=All>
            UPDIND=...;         Update Indicator:
                                <Y=Yes,
                                  N=No>
            CTC=...;            Control code:
                                <I=Inquiry,
                                  C=Change>
        }                       End CTL aggregate
        ACL{                    Action line {1,2}
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
            ACT=...;                Action code:
                                    <O=Old,
                                      N=New,
                                      R=Replace>
            NETYP=...;              Network element type
            NEGEN=...;              Network element generic
            FGRP=...;               Feature group tag
            FTR=...;                Feature tag
            FGRP{                   Feature Group Data
                SERVICE=...;        Service Type {1-3}:
                                    <B=Business,
                                      R=Residence,
                                      C=Centrex
            }                       End FGRP aggregate
            FEATURE{                Feature Data
                ACTIND=...;         Activation indicator
                ELACTIND=...;       Electronic activation indicator
                REGSET=...;         Regular set
                FTRSCP=...;         Feature scope
                SERVICE=...;        <B=Business,
                                      R=Residence,
                                      C=Centrex
            }                       End FEATURE aggregate
        }                           End ACL aggregate
    }                               End REC aggregate
}%                                  End *RCAT section
```

The *RCAT section contains a CTL aggregate and one or more ACL aggregates. The CTL aggregate identifies certain control information. In particular, the control code (CTC) is used to determine the type of control action to be performed:

- Inquire about catalog data (CTC=I)
- Change catalog data (CTC=C).

The catalog option indicator (CATOPT) determines the type of information to work with:

- Feature Group Catalog Data (CATOPT=G)
- Feature Catalog Data (CATOPT=F).

The propagate option (PROP) determines the generic to which the changes apply. If PROP=ALL, then all generics should change without specifying a generic tag. If PROP = Y, propagate the changes to the specified generic and all higher generics. If PROP = N, then only apply changes to the specified generic. Note that the PROP tag applies only to a change code (CTC = C) and not an inquiry.

The UPDIND is used for a confirmation of an update. If services are changed on the catalog, revalidation of feature exception data must be done. If specific features and services fail the validation, then that data is returned in the XCPDATA of the response section, along with a processing code message. The message questions if the update should still be done even though there were feature exceptions that failed validations. The UPDIND tag specifies "Y" if the update should be done and implies a delete of any feature exception that failed revalidation. Note that this indicator is used in conjunction with the feature option only (CATOPT = F).

The ACL aggregate contains the specific information required to perform the feature group or feature group catalog inquiry or update. The information required in each ACL aggregate varies, depending upon the CTC and CATOPT tags that appear in the CTL aggregate.

The following subsections explain how the *RCAT section should be constructed, depending upon the values in the CTL aggregate.

6.2.2.2 *RCAT Section Format: Catalog Data Inquiry (INQCAT)

An inquiry for catalog data requires at least the switch type and switch generic tags. If CATOPT = G, indicating a request for feature group information, then using NETYP and NEGEN returns multiple feature groups and related service information including description names for that specified type and generic. If FGRP is also supplied, a specific single feature group and related service information is returned.

If CATOPT = F, indicating a request for feature catalog information, then using NETYP and NEGEN returns all the features and related information for that type and generic, including the group tag IDs and descriptions of those features. If FGRP is supplied, then all the features and related information for that specific feature group are returned. If FTR is supplied,† then that specific feature with related information only is returned.

† Note if the FTR tag is supplied, the FGRP tag does not have to be specified.

The following data items may appear in the *RCAT section:

```
*RCAT{                   Request for Catalog Data
    REC{                 Record {1-N}, where N equals the number of
                         Catalog Data Requests
        CTL{             Control data
            CTC=I;       Control Code = Inquiry
            CATOPT=G;    Catalog Option = Group level
        }                End CTL aggregate
        ACL{             Action line
            NETYP=...;   Network element type
            NEGEN=...;   Network element generic
            FGRP=...;    Feature group tag [opt]
        }                End ACL aggregate
    }                    End REC aggregate
}%                       End *RCAT section
```

```
*RCAT{                   Request for Catalog Data
    REC{                 Record {1-N}, where N equals the number of
                         Catalog Data Requests
        CTL{             Control data
            CTC=I;       Control Code = Inquiry
            CATOPT=F;    Catalog Option = Feature level
        }                End CTL aggregate
        ACL{             Action line
            NETYP=...;   Network element type
            NEGEN=...;   Network element generic
            FGRP=...;    Feature group tag [opt]
            FTR=...;     Feature tag [opt]
        }                End ACL aggregate
    }                    End REC aggregate
}%                       End *RCAT section
```

6.2.2.3 *RCAT Section Format: Change Catalog Data (MTCCAT)

There are two types of changes (CTC = C) allowed for catalog data. The first provides an old and new ACL with action codes (ACT) of "O" and "N". The old ACL holds old key values and old tag values, reflecting the current image of the data, while the new ACL holds the new values allowing the old data to be changed to the new data. For the case of optional tags, *the lack of a tag in the old ACL implies the value becomes its default value (0 for integer, space for* *character).* Also for optional tags, *the lack of a tag in the new ACL implies that the value of the tag is set to its default value (0 for integer, space for character).* Comparing the old values ensures proper sequencing of the FCIF messages.

The second type of change contains one ACL with an action code of "R" for replace. The old keys are given with the tags that contain the new values only. To remove an existing value without assigning a new value, it must be explicitly assigned, dependent upon its data type. This is because new tag values only are given with this type of change. The assignment for integers is "TAG = 0;". Characters are assigned "TAG = ;". This type of change is flexible since it allows the ability to change fields without prior knowledge of their old values. This method does not support any updates of key fields.

6.2.2.3.1 *RCAT Section Format: Change Catalog Data (MTCCAT) ACT = O/N

Although the use of old and new ACLs would normally allow changes to key fields, it is restricted in the case of catalog data since Bellcore provides the data. † Therefore, in the case of CATOPT = G, the tags NETYP, NEGEN, and FGRP in the old ACL must match the values in the new ACL. In the case of CATOPT = F, the tags NETYP, NEGEN, and FTR in the old ACL must match the values in the new ACL. In this last case, the FGRP tag is not needed because a feature may only belong to one group. In addition, in the examples shown below, the lack of a service type would mean that the specific service is not available.

The following data items‡ may appear in the *RCAT section:

```
*RCAT{                      Request for Catalog Data
    REC{                    Record {1-N}, where N equals the number of
                            Catalog Data Requests
        CTL{                Control data
            CATOPT=G;       Catalog option - Group level
            PROP=Y;         Propagation - Yes [opt]
            CTC=C;          Control code - Change
```

† Service information and other feature-related indicators may be changed for particular features or groups within a type or generic, but the keys may not be changed.

‡ Maintenance for name descriptions is not done in this section. This appears in the *RDES section.

```
        }                          End CTL aggregate
        ACL{                       Action line
            ACT=O;                 Action Code - old
            NETYP=...;             Network element type
            NEGEN=...;             Network element generic
            FGRP=...;              Feature group tag
            FGRP{                  Feature Group aggregate
                SERVICE=...;       Service type {0-3} [opt]
            }                      End Feature Group aggregate
        }                          End ACL aggregate
        ACL{                       Action line
            ACT=N;                 Action Code - new
            NETYP=...;             Network element type
            NEGEN=...;             Network element generic
            FGRP=...;              Feature group tag
            FGRP{                  Feature Group aggregate
                SERVICE=...;       Service type {0-3} [opt]
            }                      End Feature Group aggregate
        }                          End ACL aggregate
    }                              End REC aggregate
}%                                 End *RCAT section

*RCAT{                               Request for Catalog Data
    REC{                             Record {1-N}, where N equals the number of
                                     Catalog Data Requests
        CTL{                         Control data
            CATOPT=F;                Catalog option - Feature level
            PROP=N;                  Propagation - No [opt]
            UPDIND=...;              Update Indicator [opt]
            CTC=C;                   Control code - Change
        }                            End CTL aggregate
        ACL{                         Action line
            ACT=O;                   Action Code - old
            NETYP=...;               Network element type
            NEGEN=...;               Network element generic
            FTR=...;                 Feature tag
            FEATURE{                 FEATURE aggregate
                SERVICE=...;         Service type {0-3} [opt]
                ACTIND=...;          Activation indicator
                ELACTIND=...;         Electronic activation indicator
                REGSET=...;           Regular set
                FTRSCP=...;           Feature scope
            }                         End Feature aggregate
        }                             End ACL aggregate
        ACL{                          Action line
            ACT=N;                    Action Code - new
            NETYP=...;                Network element type
            NEGEN=...;                Network element generic
            FTR=...;                  Feature tag
            FEATURE{                  Feature aggregate
                SERVICE=...;          Service type {0-3} [opt]
                ACTIND=...;           Activation indicator
                ELACTIND=...;         Electronic activation indicator
                REGSET=...;           Regular set
                FTRSCP=...;           Feature scope
            }                         End Feature aggregate
        }                             End ACL aggregate
    }                                 End REC aggregate
}%                                    End *RCAT section
```

6.2.2.3.2 *RCAT Section Format: Change Feature Data (MTCCAT) ACT=RPL

In this case, a CTC of "C" and only one ACL aggregate are required. For CATOPT=G, only the services for a specific group may be changed. For CATOPT=F, only the services and feature-related information may be changed for a specific feature.

The following data items may appear in the *RCAT section:

```
*RCAT{                            Request for Catalog Data
    REC{                          Record {1-N}, where N equals the number of
                                  Catalog Data Requests
        CTL{                      Control data
            CTC=C;                Control code - Change
            CATOPT=G;             Catalog Option - Group level
            PROP=A;               Propagation - All
        }                         End CTL aggregate
        ACL{                      Action line
            ACT=R;                Action code - replace
```

```
          NETYP=...;         Network element type
          NEGEN=...;         Network element generic
          FGRP=...;          Feature Group tag
          FGRP{              Feature Group aggregate
              SERVICE=...;   Service type {0-3} [opt]
          }                  End FGRP aggregate
      }                      End ACL aggregate
  }                          End REC aggregate
}%                           End *RCAT section

*RCAT{                         Request for Catalog Data
    REC{                       Record {1-N}, where N equals the number of
                               Catalog Data Requests
        CTL{                   Control data
            CTC=C;             Control code - Change data
            PROP=N;            Propagation - No
            UPDIND=...;        Update Indicator [opt]
            CATOPT=F;          Catalog Option - Feature level
        }                      End CTL aggregate
        ACL{                   Action line
            ACT=R;             Action code - replace
            NETYP=...;         Network element type
            NEGEN=...;         Network element generic
            FTR=...;           Feature tag
            FEATURE{           Feature aggregate
                SERVICE=...;   Service type {0-3} [opt]
                ACTIND=...;    Activation indicator [opt]
                ELACTIND=...;  Electronic activation indicator [opt]
                REGSET=...;    Regular set [opt]
                FTRSCP=...;    Feature scope [opt]
            }                  End FEATURE aggregate
        }                      End ACL aggregate
    }                          End REC aggregate
}%                             End *RCAT section
```

6.2.3 Request Name Description Data (*RDES) Section

The *RDES section contains the data needed to perform inquiries and maintenance to feature and feature group descriptions. Name descriptions appear through normal inquiry of catalog information in the *RCAT section.

6.2.3.1 General *RDES Section Format

The data items† for the *RDES section are:

```
*RDES{                          Request for Name Description Data
   REC{                         Record {1-N}, where N equals the number of
                                Name Description Data Requests
      CTL{                      Control data
         CTC=...;               Control code:
                                <I=Inquiry,
                                  C=Change>
      }                         End CTL aggregate
      ACL{                      Action line {1,2}
         ACT=...;               Action code:
                                <O=Old,
                                  N=New,
                                  R=Replace>
         NETYP=...;             Network element type
         TAGTYP=...;            Tag Type:
                                <F=Feature,
                                  G=Group>
         TAGID=...;             Tag ID
         DESCR{                 Description Data {1-N}
            SERVICE=...;        Service Type:
                                <B=Business,
                                  R=Residence,
                                  C=Centrex
               DESC=...;        Name Description
               PROD=...;        Product Name
         }                      End DESCR aggregate
      }                         End ACL aggregate
   }                            End REC aggregate
}%                              End *RDES section
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

The *RDES section contains a CTL aggregate and one or more ACL aggregates. The CTL aggregate identifies certain control information. In particular, the control code (CTC) is used to determine the type of control action to be performed:

- Inquire about name description data (CTC = I)
- Change name description data (CTC = C).

The following subsections explain how the *RDES section should be constructed, depending upon the CTC value in the CTL aggregate.

6.2.3.2 *RDES Section Format: Name Description Data Inquiry (INQCAT)

An inquiry for name description data requires the NETYP and TAGTYP tags. In this case, all the name descriptions are returned for every tag ID. If a particular tag ID is specified (TAGID), then all the name descriptions are returned for that particular tag ID.

The following data items may appear in the *RDES section:

```
*RDES{                        Request for Name Description Data
    REC{                      Record {1-N}, where N equals the number of
                              Name Description Data Requests
        CTL{                  Control data
            CTC=I;            Control Code = Inquiry
        }                     End CTL aggregate
        ACL{                  Action line
            NETYP=...;        Network element type
            TAGTYP=...;       Tag Type
            TAGID=...;        Tag  ID [opt]
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RDES section
```

6.2.3.3 *RDES Section Format: Change Name Description Data (MTCCAT)

There are two types of changes allowed for Name Description Data. See the description of changing catalog data for the *RCAT section for more information.

6.2.3.3.1 *RDES Section Format: Change Name Description Data (MTCCAT) ACT=O/N

This type of change requires a CTC of "C" and two ACL aggregates. Although this case would normally support key changes, there is a restriction here because the data is provided by Bellcore. Therefore, the NETYP, TAGTYP, and

TAGID tags may not be changed, which means that the old and new values for these tags must be the same.

The following data items may appear in the *RDES section:

```
*RDES{                        Request for Name Description Data
    REC{                      Record {1-N}, where N equals the number of
                              Name Description Data Requests
        CTL{                  Control data
            CTC=C;            Control Code - Change
        }                     End CTL aggregate
        ACL{                  Action line
            ACT=O;            Activity code - Old
            NETYP=...;        Network element type
            TAGTYP=...;       Tag Type
            TAGID=...;        Tag  ID
            DESCR{            Description Data {1-N}
                SERVICE=...;  Service Type [conditional]
                DESC=...;     Name Description
                PROD=...;     Product Name [conditional]
            }                 End DESCR aggregate
        }                     End ACL aggregate
        ACL{                  Action line
            ACT=N;            Action code - New
            NETYP=...;        Network element type
            TAGTYP=...;       Tag Type
            TAGID=...;        Tag  ID
            DESCR{            Description Data {1-N}
                SERVICE=...;  Service Type [conditional]
                DESC=...;     Name Description
                PROD=...;     Product Name [conditional]
            }                 End DESCR aggregate
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RDES section
```

6.2.3.3.2 *RDES Section Format Change Name Description Data (MTCCAT) ACT=RPL

In this type of change, a CTC of "C" and one ACL aggregate are required. The NETYP, TAGTYP, TAGID, and SERVICE tags must be specified in order to use this type of change.

The following data items may appear in the *RDES section:

```
*RDES{                          Request for Catalog Data
   REC{                         Record {1-N}, where N equals the number of
                                Name Description Data Requests
      CTL{                      Control data
         CTC=C;                 Control Code = Change
      }                         End CTL aggregate
      ACL{                      Action line
         ACT=R;                 Activity code - replace
         NETYP=...;             Network element type
         TAGTYP=...;            Tag Type
         TAGID=...;             Tag ID
         DESCR{                 Description Data {1-4}
            SERVICE=...;        Service Type [opt]
            DESC=...;           Name Description [opt]
            PROD=...;           Product Name [opt]
         }                      End DESCR aggregate
      }                         End ACL aggregate
   }                            End REC aggregate
}%                              End *RDES section
```

6.2.4 Catalog Related Data Response (*CATR) Section

The *CATR section is created as a result of a request in the *RCAT section of the INQCAT or MTCCAT contract. There is one response REC for each REC in the original request message. The section contains responses to feature group catalog and feature catalog information. The *CATR section also identifies errors and discrepancies.

6.2.4.1 General *CATR Section Format

The data items† for the *CATR section are shown below in a general format. The *RCAT of the request message is provided in the CTL and ACL aggregates of the corresponding *CATR section of the response message for each REC

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

aggregate. The items shown in regular typeface reflect the data provided in the original request (*RCAT section). The items shown in bold typeface represent the response data.

```
*CATR{                          Catalog Data Response
   REC{                         Record {1-N}, where N equals the number of
                                Catalog Data Responses
    CTL{                        Control data
        CATOPT=...;             Catalog option
        PROP=...;               Propagation option
        UPDIND=...;             Update indicator
        CTC=...;                Control code
        IERC=...;               Input error return code {0-N}, where N=the
                                number of input errors
        CDRC=...;               Catalog data return code {0-N}
    }                           End CTL aggregate
    ACL{                        Action line {1,2}
          NETYP=...;            Network element type
          NEGEN=...;            Network element generic
          FGRP=...;             Feature group tag
          FTR=...;              Feature tag
          FGRP{                 FGRP aggregate
              SERVICE=...;      Service Type {0-3}
          }                     End FGRP aggregate
          FEATURE{              Feature Data Information
              SERVICE=...;      Service Type {0-3}
              ACTIND=...;       Activation indicator
              ELACTIND=...;     Electronic activation indicator
              REGSET=...;       Regular set
              FTRSCP=...;       Feature scope
          }                     End FEATURE aggregate
    }                           End ACL aggregate
    FGRPDATA{                   Feature group information {0-N}
        FGRP=...;               Feature group tag ID
        GDESCR{                 GDESCR aggregate {0-3}
            SERVICE=...;        Service Type
            FGRPDES=...;        Feature Group Name Description
        }                       End GDESCR aggregate
    }                           End FGRPDATA aggregate
```

```
        FEATDATA{                       Feature data information {0-N}
            FGRP=...;                   Feature group tag ID
            GDESCR{                     GDESCR aggregate {0-3}
                SERVICE=...;            Service Type
                FGRPDES=...;            Feature Group Name Description
            }                           End GDESCR aggregate
            FTR=...;                    Feature tag ID
            FDESC{                      FDESC aggregate {0-3}
                SERVICE=...;            Service Type
                FTRDES=...;             Feature Name Description
            }                           End FDESC aggregate
            ACTIND=...;                 Activation indicator
            ELACTIND=...;               Electronic activation indicator
            REGSET=...;                 Regular set
            FTRSCP=...;                 Feature scope
        }                               End FEATDATA aggregate
        XCPDATA{                        Switch feature exception data {0-N}
            FTR=...;                    Feature tag
            SID{                        SID aggregate (1-N}
                CLLI=...;               Network element ID
                SERVICE=...;            Service Type
            }                           End SID aggregate
        }                               End XCPDATA aggregate
        ERROR{                          ERROR aggregate {0-N}
            SERVICE...;                 Service type
            NEGEN...;                   Network element generic
            SVCERR...;                  Service error code
        }                               End ERROR aggregate
    }                                   End REC aggregate
}%                                      End *CATR section
```

The first group of response data (IERC and CDRC), includes the CTL aggregate return codes. This data is returned as a response in all cases (CTC = I, C).

The other group of response data (FGRPDATA and/or FEATDATA aggregates) shows the additional data that may appear in the *CATR section. This is returned upon a successful inquiry request only.

The XCPDATA aggregate is returned only when a change of service leads to a failure in revalidating the feature exceptions against the new catalog containing the updated services. A CDRC processing message supplies the failed feature exception validations and the UPDIND tag to determine if the update should continue. See the description of the *RCAT section for an explanation of the UPDIND tag.

The ERROR aggregate is returned when there is an error. The aggregate identifies the generic that is affected for all IERC errors. This is done specifically for the case in which the PROP tag specifies that multiple generics are being manipulated. In the case of a service type error, the service type and associated generic is identified. This is done because several service types could be input at the same time.

The return codes and the additional aggregates are explained in the sections that follow the general format.

6.2.4.2 CTL Aggregate Return Codes

There are two return code tags that may be included in the CTL aggregate. In the general format above, they are shown in bold typeface within the CTL aggregate. These return code tags are:

- Input Error Return Code (IERC)

  This return code indicates an incorrect format in the FCIF message or the associated tags. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors. See Appendix D for specific IERC values and descriptions.

- Feature Data Return Code (CDRC)

  The CDRC tag indicates the result of inquiries, and changes of feature related data. This tag may appear multiple times indicating multiple process messages. See Appendix D for specific CDRC values and descriptions.

6.2.4.3 *CATR Section Format: Feature Data Inquiry Response

Upon successful completion of an inquiry, the data items in the FGRPDATA and/or FEATDATA aggregate of the *CATR general section are returned (depending upon the CATOPT specified).

If unsuccessful, only the IERC and CDRC tags are returned.

6.2.4.4 *CATR Section Format: MTCCAT Contract Response

The response to maintenance activity (CTC = C) results in either a successful or unsuccessful return code, but it does not return any data as would happen in the case of an inquiry request. Therefore, for this case, the FGRPDATA and/or FEATDATA aggregates are never returned. Only the IERC and CDRC tags are returned, which contain the input error codes and return codes described previously.

```
*CATR{                        Catalog Data Response
    REC{                      Record {1-N}, where N equals the number of
                              Catalog Data Responses
     CTL{                     Control data
          CTC=...;            Control code <C>
          IERC=...;           Input error return code {0-N}, where N= the
                              number of input errors
          CDRC=...;           Catalog data return code {0-N}
      }                       End CTL aggregate
      ACL{                    Action line {1,2} from original request
      }                       End ACL aggregate
    }                         End REC aggregate
}%                            End *CATR section
```

6.2.5 Name Description Data Response (*DESR) Section

The *DESR section is created as a result of a request in the *RDES section of the INQCAT or MTCCAT contract. There is one response REC for each REC in the original request message. The section contains responses to name description data. The *DESR section also identifies errors and discrepancies.

6.2.5.1 General *DESR Section Format

The data items† for the *DESR section are shown below in a general format. The *RDES of the request message is provided in the CTL and ACL aggregates of the corresponding *DESR section of the response message for each REC aggregate. The items shown in regular typeface reflect the data provided in the original request (*RDES section). The items shown in bold typeface represent the response data.

```
*DESR{                            Name Description Data Response
    REC{                          Record {1-N}, where N equals the number of
                                  Name Description Data Responses
     CTL{                         Control data
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
            CTC=...;               Control code
            IERC=...;              Input error return code {0-N}, where N=the
                                   number of input errors
            DDRC=...;              Description data return code {0-N}
        }                          End CTL aggregate
        ACL{                       Action line {1,2}
              NETYP=...;           Network element type
              TAGTYP=...;          Tag Type
              TAGID=...;           Tag ID
              DESCR{               DESCR aggregate
                  SERVICE=...;     Service Type
                  DESC=...;        Name Description
                  PROD=...;        Product Name
              }                    End DESCR aggregate
        }                          End ACL aggregate
        NAMES{                     NAMES aggregate {0-N}
              TAGID=...;           Tag ID
              DESCR{               DESCR aggregate {1-N}
                  SERVICE=...;     Service Type
                  DESC=...;        Name Description
                  PROD=...;        Product Name
              }                    End DESCR aggregate
              ERROR{               ERROR aggregate {0-N}
                  SERVICE=...;     Service Type
                  DESC=...;        Name Description
                  PROD=...;        Product Name
                  NAMERR=...;      Names Error Code
              }                    End ERROR Aggregate
         }                         End NAMES aggregate
     }                             End REC aggregate
}%                                 End *DESR section
```

The first group of response data (IERC and DDRC) includes the CTL aggregate return codes. This is returned as a response in all cases (CTC = I, C).

The other group of response data (the DESCR aggregate) shows the additional data that may appear in the *CATR section. This is returned only upon a successful inquiry request. The return codes and the additional aggregates are explained in the sections that follow the general format.

The ERROR aggregate is returned when any errors occur. This aggregate identifies the data fields in error and provides the corresponding error code.

6.2.5.2 CTL Aggregate Return Codes

There are two return code tags that may be included in the CTL aggregate. In the general format above, they are shown in bold typeface within the CTL aggregate. These return code tags are:

- Input Error Return Code (IERC)

  This return code indicates an incorrect format in the FCIF message or the associated tags. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors. See Appendix D for specific IERC values and descriptions.

- Description Data Return Code (DDRC)

  The DDRC tag indicates the result of inquiries and changes of feature-related data. This tag may appear multiple times indicating multiple process messages. See Appendix D for specific DDRC values and descriptions.

6.2.5.3 *DESR Section Format: Name Description Data Inquiry Response

Upon successful completion of an inquiry, the data items in the DESCR aggregate of the *DESR general section are returned.

If unsuccessful, only the IERC and DDRC tags are returned.

6.2.5.4 *DESR Section Format: MTCCAT Contract Response

The response to maintenance activity (CTC = C) results in either a successful or unsuccessful return code, but does not return any data as would happen in the case of an inquiry request. Therefore, for this case, the NAMES aggregate is never returned. Only the IERC and DDRC tags are returned. These tags hold the input errors and return codes described previously.

```
*DESR{                        Name Description Data Response
    REC{                      Record {1-N}, where N equals the number of
                              Name Description Data Responses
     CTL{                     Control data
          CTC=...;            Control code <C>
          IERC=...;           Input error return code {0-N}, where N=the
                              number of input errors
```

```
        DDRC=...;          Description Data return code {0-N}
      }                    End CTL aggregate
      ACL{                 Action line {1,2} from original request
      }                    End ACL aggregate
    }                      End REC aggregate
}%                         End *DESR section
```

6.3 CPE Data Maintenance Contracts

This section describes two contracts, INQCPE and MTCCPE, which are used to maintain CPE data. CPE data encompasses customer premises equipment information and reference data. The INQCPE contract supports the inquiry† of CPE data, while the MTCCPE contract supports the maintenance (build, change, and remove) of CPE data.

Both contracts consist of a request and a response message and share the *RCPE and *CPER sections listed below.

- The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section (optional)
*RCPE	Request for Customer Premises Equipment section

- The FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section (optional)
*SNSHDR	SNS Header section
*CPER	Customer Premises Equipment Response section

See Section 3 in this document for more information on the *C1, *C2, *PLHDR, *UMSG, and *SNSHDR FCIF sections. The *RCPE and the *CPER sections are explained in the following subsections.

6.3.1 General Contract Request Information

The contracts are designed so that multiple requests may be initiated in one contract invocation (request message). This is denoted by the REC aggregate that spans from {1-N}. Therefore, the INQCPE contract may contain multiple inquiry requests, while the MTCCPE contract may contain multiple requests for

† Note that the inquiries in maintenance are designed for the support of the maintenance functionality.

builds, changes, and deletes. These requests are structured in terms of the following FCIF sections. The *RCPE section supports the maintenance of CPE data:

- Inquiry of CPE Data (INQCPE)
- Build CPE Data (MTCCPE)
- Change CPE Data (MTCCPE)
- Remove CPE Data (MTCCPE).

6.3.2 Request for CPE Data (*RCPE) Section

The *RCPE section contains the data needed to perform inquiries and maintenance to CPE data.

6.3.2.1 General *RCPE Section Format

The data items[†] for the *RCPE section are:

```
*RCPE{                           Request for CPE data
  REC{                           Record {1-N}, where N equals the number
                                 of CPE Inquiry Requests
      CTL{                       Control data
          CTC=...;               Control code:
                                 <I=Inquiry,
                                   B=Build,
                                   C=Change,
                                   R=Remove>
      }                          End CTL aggregate
      ACL{                       Action line {1,2}
          ACT=...;               Action code:
                                 <O=Old,
                                   N=New,
                                   R=Replace>
          CPEMAN=...;            CPE manufacturer
```

[†] Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
                CPEMOD=...;          CPE model number
                CPENAME=...;         CPE name
                CPEREG=...;          CPE registration number
                DIS=...;             Display indicator
                ONETOUCH=...;        One touch indicator
                MAXKEY=...;          Maximum number of keys
                MAXCA=...;           Maximum number of call appearances
                HKEY=...;            Hold key indicator
                RKEY=...;            Release key indicator
                NETYP=...;           Switch type {0-N}
            }                        End ACL aggregate
    }                                End REC aggregate
}%                                   End *RCPE section
```

The *RCPE section contains a CTL aggregate and one or more ACL aggregates. The CTL aggregate identifies certain control information. In particular, the control code (CTC) is used to determine the type of control action to be performed:

- Inquire about CPE data (CTC = I)
- Add/Build CPE data (CTC = B)
- Change CPE data (CTC = C)
- Remove CPE data (CTC = R).

The ACL aggregate contains the specific information required to perform the CPE data inquiry or update. The information required in each ACL aggregate varies, depending upon the CTC that appears in the CTL aggregate.

The following subsections explain how the *RCPE section should be constructed, depending upon the CTC value in the CTL aggregate.

6.3.2.2 *RCPE Section Format: Customer Premises Equipment Inquiry (INQCPE)

An inquiry for CPE data requires the CPEMAN and CPEMOD (manufacturer and model number) tags.

The following data items may appear in the *RCPE section:

```
*RCPE{                        Request for CPE data
    REC{                      Record {1-N}, where N equals the number
                              of CPE Inquiry Requests
        CTL{                  Control data
            CTC=I;
        }                     End CTL aggregate
        ACL{                  Action line
            CPEMAN=...;       CPE manufacturer
            CPEMOD=...;       CPE model number
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RCPE section
```

6.3.2.3 *RCPE Section Format: Build CPE Data (MTCCPE)

To add CPE data† requires a CTC of "B" and an ACT of "N". The required tags in the ACL aggregate are CPEMAN, CPEMOD, DIS, ONETOUCH, HKEY, RKEY and MAXKEY.

The following data items may appear in the *RCPE section:

```
*RCPE{                        Request for CPE data
    REC{                      Record {1-N}, where N equals the number
                              of CPE Maintenance Requests
        CTL{                  Control data
            CTC=B;            Control code
        }                     End CTL aggregate
        ACL{                  Action line
            ACT=N;            Action code = New
            CPEMAN=...;       CPE manufacturer
            CPEMOD=...;       CPE model number
            CPENAME=...;      CPE name [opt]
            CPEREG=...;       CPE registration number [opt]
```

† Note that multiple switch types (NETYP) may be added for a CPE manufacturer and model number.

```
                DIS=...;            Display indicator
                ONETOUCH=...;       One touch indicator
                MAXKEY=...;         Maximum number of keys
                MAXCA=...;          Maximum number of call appearances [opt]
                HKEY=...;           Hold key indicator
                RKEY=...;           Release key indicator
                NETYP=...;          Switch type {0-N} [opt]
            }                       End ACL aggregate
        }                           End REC aggregate
}%                                  End *RCPE section
```

6.3.2.4 *RCPE Section Format: Remove CPE Data (MTCCPE)

To delete CPE data, a CTC of "R" and an ACT of "O" are required in the CTL aggregate. In the ACL aggregate, the CPEMAN and CPEMOD tags are required.

The following data items may appear in the *RCPE section:

```
*RCPE{                          Request for CPE data
    REC{                        Record {1-N}, where N equals the number
                                of CPE Maintenance Requests
        CTL{                    Control data
            CTC=R;              Control code - Remove
        }                       End CTL Aggregate
        ACL{                    Action line
            ACT=O;              Action code - Old
            CPEMAN=...;         CPE manufacturer
            CPEMOD=...;         CPE model number
        }                       End ACL aggregate
    }                           End REC aggregate
}%                              End *RCPE section
```

6.3.2.5 *RCPE Section Format: Change Customer Premises Equipment (MTCCPE)

There are two types of changes (CTC = C) allowed for CPE data. The first provides an old and new ACL with action codes (ACT) of "O" and "N". This allows updates for key fields. The old ACL holds old key values and old tag values that reflect the current data image, while the new ACL holds the new values that allow the old data to be changed to the new data. *The lack of a tag in the old ACL implies that the value represents its default value (0 for integer, or space for character). The lack of a tag in the new ACL implies that the value of the tag will be set to its default value.* This case is flexible since it allows key changes and ensures proper sequencing by comparing tag values in the old ACL to the old database image.

The second type of change contains one ACL with an action code of "R" for replace. The old keys are given with the tags that contain the new values only. To remove an existing value without assigning a new value, it must be explicitly assigned, dependent upon its data type. The assignment for integers is TAG=0; while characters will be assigned TAG= ;. This type of change is flexible since it allows the ability to change fields without prior knowledge of their old values. However, this method does not allow changes to key fields.

6.3.2.5.1 *RCPE Section Format: Change Customer Premises Equipment (MTCCPE) ACT=O/N

The following data items may appear in the *RCPE section:

```
*RCPE{                      Request for CPE data
    REC{                    Record {1-N}, where N equals the number
                            of CPE Maintenance Requests
        CTL{                Control data
            CTC=C;          Control code - Change
        }                   End CTL aggregate
        ACL{                Action line
            ACT=O;          Action code - Old
            CPEMAN=...;     CPE manufacturer
            CPEMOD=...;     CPE model number
            CPENAME=...;    CPE name [opt]
            CPEREG=...;     CPE registration number [opt]
            DIS=...;        Display indicator
            ONETOUCH=...;   One touch indicator
            MAXKEY=...;     Maximum number of keys
            MAXCA=...;      Maximum number of call appearances [opt]
            HKEY=...;       Hold key number
            RKEY=...;       Release key number
            NETYP=...;      Switch type {0-N} [opt]
        }                   End ACL aggregate
        ACL{                Action line
            ACT=N;          Action code - New
            CPENAME=...;    CPE name [opt]
            CPEREG=...;     CPE registration number [opt]
            CPEMOD=...;     CPE model number
```

```
          DIS=...;           Display indicator
          ONETOUCH=...;      One touch indicator
          MAXKEY=...;        Max number of keys
          MAXCA=...;         Maximum number of call appearances [opt]
          HKEY=...;          Hold key indicator
          RKEY=...;          Release key indicator
          NETYP=...;         Switch type {0-N} [opt]
      }                      End ACL aggregate
   }                         End REC aggregate
}%                           End *RCPE section
```

6.3.2.5.2 *RCPE Section Format: Change Customer Premises Equipment (MTCCPE) ACT=RPL

In this type of change, a CTC of "C" and one ACL aggregate only are required. As a reminder, you cannot change the key field with this type of change.

The following data items may appear in the *RCPE section:

```
*RCPE{                       Request for CPE data
   REC{                      Record {1-N}, where N equals the number
                             of CPE Maintenance Requests
      CTL{                   Control data
          CTC=C;             Control code - Change
      }                      End CTL aggregate
      ACL{                   Action line
          ACT=R;             Action code - replace
          CPEMAN=...;        CPE manufacturer
          CPEMOD=...;        CPE model number
          CPENAME=...;       CPE name [opt]
          CPEREG=...;        CPE registration number [opt]
          DIS=...;           Display indicator
          ONETOUCH=...;      One touch indicator
          MAXKEY=...;        Maximum number of keys
          MAXCA=...;         Maximum number of call appearances [opt]
          HKEY=...;          Hold key indicator
```

```
            RKEY=...;              Release key indicator
            NETYP=...;             Switch type {0-N} [opt]
        }                          End ACL aggregate
    }                              End REC aggregate
}%                                 End *RCPE section
```

6.3.3 Customer Premise Equipment Response (*CPER) Section

The *CPER section is created as a result of a request in the *RCPE section of the INQCPE or MTCCPE contract. There is one response REC for each REC in the original request message. The section contains responses to CPE data requests. The *CPER section also identifies errors and discrepancies.

6.3.3.1 General *CPER Section Format

The data items† for the *CPER section are shown below in a general format. The *RCPE section of the request message is provided in the CTL and ACL aggregates of the corresponding *CPER section of the response message for each REC aggregate. The items shown in regular typeface reflect the data provided in the original request (*RCPE section). The items shown in bold typeface represent the response data.

```
*CPER{                             Customer Premises Equipment Response
    REC{                           Record {1-N}, where N equals the number
                                   of Customer Premises Equipment
                                   Responses
     CTL{                          Control data
          CTC=...;                 Control code
          IERC=...;                Input error return code {0-N}, where
                                   N=the number of input errors
          CPRC=...;                Customer Premises Equipment return
                                   code {0-N}
      }                            End CTL aggregate
      ACL{                         Action line {1,2}
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
                CPEMAN=...;        CPE manufacturer
                CPEMOD=...;        CPE model number
                CPENAME=...;       CPE name
                CPEREG=...;        CPE registration number
                DIS=...;           Display indicator
                ONETOUCH=...;      One touch indicator
                MAXKEY=...;        Maximum number of keys
                MAXCA=...;         Maximum number of call appearances
                HKEY=...;          Hold key indicator
                RKEY=...;          Release key indicator
                NETYP=...;         Switch type {0-N}
            }                      End ACL aggregate
            CPEDATA{               Customer Premises Equipment data {0-1}
                CPENAME=...;       CPE name
                CPEREG=...;        CPE registration number
                DIS=...;           Display indicator
                ONETOUCH=...;      One touch indicator
                MAXKEY=...;        Maximum number of keys
                MAXCA=...;         Maximum number of call appearances
                HKEY=...;          Hold key indicator
                RKEY=...;          Release key indicator
                NETYP=...;         Network element type {0-N}
            }                      End CPEDATA aggregate
          NETYPERR{                NETYPERR aggregate {0-N}
                NETYP=...;         Network element type
                NERR=...;          Network element error
            }                      End NETYPERR aggregate
      }                            End REC aggregate
}%                                 End *CPER section
```

The first group of response data (IERC and CPRC) includes the CTL aggregate return codes. This data is returned as a response in all cases (CTC = I, B, C, R).

The CPEDATA aggregate shows the additional data that may appear in the *CPER section. This is returned only upon a successful inquiry request.

If an error is detected with the input NETYP data, the NETYPERR aggregate will be generated on output. This is done to indicate which specific occurrence of NETYP is in error, since several could be present on input.

The return codes and the additional aggregates are explained in the subsections below.

6.3.3.2 CTL Aggregate Return Codes

There are two return code tags that may be included in the CTL aggregate. In the general format above, they are shown in bold typeface within the CTL aggregate. These return code tags are:

- Input Error Return Code (IERC)

  This return code indicates an incorrect format in the FCIF message or the associated tags. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors. See Appendix D for specific IERC values and descriptions.

- Customer Premises Equipment Return Code (CPRC)

  The CPRC tag indicates the result of inquiries, additions, changes, replacements, and deletions of CPE data. See Appendix D for specific CPRC values and descriptions.

6.3.3.3 *CPER Section Format: Customer Premises Equipment Inquiry Response

Upon successful completion of an inquiry, the data items in the CPEDATA aggregate of the *CPER general section are returned. If unsuccessful, only the IERC and CPRC tags are returned.

6.3.3.4 *CPER Section Format: MTCCPE Contract Response

The response to any maintenance activity (CTC=B,C,R) results in either a successful or unsuccessful return code. Data is not returned as in the case of an inquiry request. Therefore, for this case, the CPEDATA aggregate is never returned. Only the IERC and CPRC tags are returned. These tags hold the input errors and return codes described previously.

```
*CPER{                        Customer Premises Equipment Response
    REC                       Record {1-N}, where N equals the number
                              of CPE Responses
      CTL{                    Control data
          CTC=...;            Control code <B,C,R>
          IERC=...;           Input error return code {0-N}, where
                              N=the number of input errors
          CPRC=...;           Customer Premises Equipment return
                              code {0-1}
      }                       End CTL aggregate
      NETYPERR{                 NETYPERR aggregate {0-N}
         NETYP=...;             Network element type
         NERR=...;              Network element error
      }                         End NETYPERR aggregate
   }                            End REC aggregate
}%                              End *CPER section
```

If an error is detected with the input NETYP data, the NETYPERR aggregate will be generated on output. This is done to indicate which specific occurrence of NETYP is in error, since several could be present on input.

6.4 Feature Package Data Maintenance Contracts

This section describes two contracts, INQFPK and MTCFPK, which are used to maintain feature package data. The INQFPK contract supports the inquiry of feature package data, while the MTCFPK contract supports the maintenance of feature package data. Both contracts consist of a request message and a response message.

FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section (optional)
*RFPK	Request for Feature Package Maintenance

FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section (optional)
*SNSHDR	SNS Header section
*FPKR	Feature Package Maintenance Response

Both contracts, INQFPK and MTCFPK, follow this FCIF format and share the *RFPK and *FPKR sections.

See Section 3 in this document for more information on the *C1, *C2, *PLHDR, *UMSG, and *SNSHDR FCIF sections.

6.4.1 Request Feature Package Data (*RFPK)

The *RFPK section allows you to build, remove, change, and inquire about feature packages. The general format of this section is shown below.

6.4.1.1 General Format

```
*RFPK{              Request for Feature Package Data
   REC{             Record {1-N}, where N equals the number
                    of Feature Package Requests
      CTL{          Control data
```

```
            CTC=...;            Control code: <I=Inquiry, B=Build,
                                C=Change, R=Remove>
            PROP=...;           Propagate Indicator
            ALL=...;            Request all data Indicator; <Y=Yes>
        }                       End CTL aggregate
        ACL{                    Action line {1,2}
            ACT=...;            Action code:
                                <O=Old,
                                N=New,
                                R=Replace>
            PROD=...;           Product Name
            NETYP=...;          Switch Type
            NEGEN=...;          Switch Generic
            FPKG=...;           Feature Package Name
            SERVICE=...;        Types of service {0-3}:
                                <R=Residence,
                                B=Business,
                                C=Centrex>
            FEATURE{            Feature {1-N} where N equals the number
                                of features within this package.
                FTR=...;        Feature Tag ID
                OPT=...;        Optional Indicator
            }                   End Feature aggregate
        }                       End ACL aggregate
    }                           End REC aggregate
}%                              End *RFPK section
```

This FCIF section is broken into the REC, CTL, ACL, and FEATURE aggregates. The REC aggregate allows multiple requests to be sent in one contract. For example, one *INQFPK contract may process multiple inquiry requests. Similarly, one *MTCFPK contract may process multiple requests for build, remove, and change. The CTL aggregate holds the control code (CTC), which determines the type of request to be performed. The ACL aggregate contains the necessary information to process this request. The FEATURE aggregate contains information about specific features in a package.

The NEGEN and PROP tags are used to determine the generic of the feature package as follows:

No NEGEN tag, PROP=A   Perform the maintenance request for all generics.

NEGEN=xyz, PROP=N   Perform the maintenance request for the specific generic xyz.

NEGEN=xyz, PROP=Y   Perform the maintenance request for the given generic xyz and all subsequent generics (propagate it).

The generic is determined in this manner for all requests in the maintenance contract, MTCFPK. In the inquiry contract, however, a specific generic must be supplied by the user.

The ALL tag is used to indicate how much data should be returned upon an inquiry request. For example, when ALL = Y, this indicates the return of both feature package data names and all the feature information associated with the feature packages. When ALL = N, this indicates that only feature package names were requested.

The following subsections describe the formats of specific requests for each control code (BRIC†). In these FCIF formats, each tag shown is required unless specifically labeled as optional, denoted by "[opt]".

6.4.1.2 *RFPK Section Format: Feature Package Inquiry (INQFPK)

There are three types of inquiries that can be made on a feature package:

1. Request a list of all feature package names under a product name/switch type/switch generic. The required tags for this type of request are PROD, NETYP, NEGEN, and ALL = N.
2. Request a list of all features under a specific feature package. The required tags for this type of request are PROD, NETYP, NEGEN, and FPKG.
3. Request all feature packages and all features within the packages. The required tags for this type of request are PROD, NETYP, NEGEN, and ALL = Y.

The absence of the feature package name tag, FPKG, indicates that all feature package names are requested. It is the ALL indicator that determines whether just feature package names or feature packages and associated feature information are returned upon an inquiry request. If ALL=Y, all feature

† BRIC is used here to abbreviate Build, Remove, Inquire and Change

packages and their associated feature information are returned. If ALL = N, feature package names only are returned. The type of service, indicated by the SERVICE tag, determines which feature package data is returned. For example, if SERVICE = "R" (residence), then residence feature package names† are returned in the response section (*FPKR). Note that only one type of service may be specified here. The lack of a SERVICE tag indicates all feature package data under that product is returned.

If the features of a specific package are requested, then FPKG should be equal to the name of the package.

To obtain a list of feature package names, the following *RFPK section would be used:

```
*RFPK{                          Request for Feature Package Data
    REC{                        Record {1-N}, where N equals the number of
                                Feature Package Requests
        CTL{                    Control data
            CTC = I;            Control code: Inquiry
        }                       End CTL aggregate
        ACL{                    Action line
            PROD = ...;         Product Name
            NETYP = ...;        Switch Type
            NEGEN = ...;        Switch Generic
            SERVICE = ...;      Type of Service   [opt]
        }                       End ACL aggregate
    }                           End REC aggregate
}%                              End *RFPK section
```

† "Residence feature package names" is used here to mean those feature packages that support residence services. These packages may also support other types of services.

To obtain a list of features for a particular feature package, the *RFPK section would appear as:

```
*RFPK{                      Request for Feature Package Data
   REC{                     Record {1-N}, where N equals the number of
                            Feature Package Requests
      CTL{                  Control data
         CTC=I;             Control code: Inquiry
      }                     End CTL aggregate
      ACL{                  Action line
         PROD=...;          Product Name
         NETYP=...;         Switch Type
         NEGEN=...;         Switch Generic
         FPKG=...;          Feature Package Name
      }                     End ACL aggregate
   }                        End REC aggregate
}%                          End *RFPK section
```

To obtain information about all feature packages and their associated features, the following *RFPK section would be used:

```
*RFPK{                      Request for Feature Package Data
   REC{                     Record {1-N}, where N equals the number of
                            Feature Package Requests
      CTL{                  Control data
         CTC=I;             Control code: Inquiry
         ALL=Y;             All indicator - Yes
      }                     End CTL aggregate
      ACL{                  Action line
         PROD=...;          Product Name
         NETYP=...;         Switch Type
         NEGEN=...;         Switch Generic
         SERVICE=...;       Service Type    [opt]
      }                     End ACL aggregate
   }                        End REC aggregate
}%                          End *RFPK section
```

6.4.1.3 *RFPK Section Format: Build Feature Package (MTCFPK)

To build a feature package, there must be at least one FEATURE aggregate in the *RFPK section.[†] The absence of the SERVICE tag implies that the feature package has been set up, however it is not yet offered (possibly because of tariff reasons). This section may appear as follows:

```
*RFPK{                     Request for Feature Package Data
   REC{                    Record {1-N}, where N equals the number of
                           Feature Package Requests
      CTL{                 Control data
         CTC=B;            Control code - Build
         PROP=...;         Propagate indicator [opt]
      }                    End CTL aggregate
      ACL{                 Action line
         ACT=N;            Action code: New data
         PROD=...;         Product Name
         NETYP=...;        Switch Type
         NEGEN=...;        Switch Generic [conditional]
         FPKG=...;         Feature Package Name
         SERVICE=...;      Type of service {0-3}
         FEATURE{          Feature {1-N} where N equals the number of
                           features within this package.
            FTR=...;       Feature Tag ID
            OPT=...;       Optional Indicator
         }                 End Feature Aggregate
      }                    End ACL aggregate
   }                       End REC aggregate
}%                         End *RFPK section
```

6.4.1.4 *RFPK Section Format: Remove Feature Package (MTCFPK)

To remove feature package data, a CTC of "R" and an ACT of "O" are required in the CTL and ACL aggregates, respectively.

The type of remove performed is determined by the appearance of the tags. Although the product name (PROD) is the only required tag, the NETYP, NEGEN, and FPKG tags may also be supplied. Depending upon the

[†] Every feature package must consist of at least one feature.

combination, various responses may result. Some of the possible combinations are:

- PROD/NETYP/NEGEN/FPKG removes a specific feature package for a switch type, switch generic, and product. The PROP tag may also be used to propagate the delete across generics.
- PROD/NETYP/NEGEN removes all feature packages for a switch type, switch generic, and product. Again, the PROP tag may also be used to propagate the delete across generics.
- PROD/FPKG removes a specific feature package from a product for all switch types and switch generics. The PROP tag is not used in this case.
- PROD removes all the feature packages for a product for all switch types and generics. The PROP tag is not used in this case.

When a feature package is removed, it is removed for all types of service (business, residence, and Centrex).

The following data items may appear in the *RFPK section:

```
*RFPK{                Request for Feature Package Data
   REC{               Record {1-N}, where N equals the number of
                      Feature Package Requests
      CTL{            Control data
      CTC=R;          Control code: Remove
      PROP=...;       Propagate indicator [opt]
      }               End CTL aggregate
      ACL{            Action line
         ACT=O;       Action code: Old data
         PROD=...;    Product Name
         NETYP=...;   Switch Type [opt]
         NEGEN=...;   Switch Generic [opt]
         FPKG=...;    Feature Package Name [opt]
      }               End ACL aggregate
   }                  End REC aggregate
}%                    End *RFPK section
```

6.4.1.5 *RFPK Section Format: Change a Feature Package (MTCFPK)

There are two ways to request a change for feature packages:

1. Old/New Action Codes

   This request requires two ACL aggregates to change the old data in the old ACL aggregate (ACT = O) to the new data supplied in the new ACL aggregate (ACT = N). The lack of a tag in the new ACL aggregate for optional fields indicates that the value of that tag becomes its default value (0 for integers, space for characters). The lack of a tag in the old ACL aggregate for optional fields indicates that its current value is its default value (0 for integers, space for characters). This type of change allows key changes and ensures proper sequencing by matching the tag values in the old ACL aggregate to the database values. PROD, NETYP, NEGEN, SERVICE, and FPKG may be changed as well as Feature Data in this type of change.

2. Replace Action Code

   This method requires one ACL aggregate (ACT = R) with an action code of replace. An old image of the data is not needed. The new data must be provided with the feature package and all of its associated features and related data. In this type of change, only feature data associated with a particular feature package for a specific product, switch type, and switch generic may be changed. Therefore, PROD, NETYP, NEGEN, and FPKG tags are required in this type of change. The propagate option (PROP) may be used to propagate changes across generics.

Using the old/new ACLs to perform the change, the *RFPK section would appear as:

```
*RFPK{                  Request for Feature Package Data
   REC{                 Record {1-N}, where N equals the number of
                        Feature Package Requests
      CTL{              Control data
         CTC = C;       Control code: Change
         PROP = ...;    Propagate indicator [opt]
      }                 End CTL aggregate
      ACL{              Action line
         ACT = O;       Action code: Old data
```

```
        PROD=...;              Product Name
        NETYP=...;             Switch Type
        NEGEN=...;             Switch Generic [conditional]
        SERVICE=...;           Type of service {0-3}
        FPKG=...;              Feature Package Name
        FEATURE{               Feature {1-N} where N equals the number of
                               features within this package.
           FTR=...;            Feature Tag ID
           OPT=...;            Optional Indicator
        }                      End Feature aggregate
     }                         End ACL aggregate
     ACL{                      Action line
        ACT=N;                 Action code: New data
        PROD=...;              Product Name
        NETYP=...;             Switch Type
        NEGEN=...;             Switch Generic [conditional]
        SERVICE=...;           Type of service {0-3}
        FPKG=...;              Feature Package Name
        FEATURE{               Feature {1-N} where N equals the number of
                               features within this package.
           FTR=...;            Feature Tag ID
           OPT=...;            Optional Indicator
        }                      End Feature aggregate
     }                         End ACL aggregate
  }                            End REC aggregate
}%                             End *RFPK section
```

Using the "R" (replace) action code, the *RFPK section for a change would appear as:

```
*RFPK{                         Request for Feature Package Data
  REC{                         Record {1-N}, where N equals the number of
                               Feature Package Requests
     CTL{                      Control data
        CTC=C;                 Control code: Change
        PROP=...;              Propagate option [opt]
     }                         End CTL aggregate
     ACL{                      Action line
```

```
            ACT=R;              Action code: Replace data
            PROD=...;           Product Name
            NETYP=...;          Switch Type
            NEGEN=...;          Switch Generic [conditional]
            FPKG=...;           Feature Package Name
            SERVICE=...;        Type of service {0-3} [opt]
            FEATURE{            Feature {0-N} where N equals the number of
                                features within this package.  Replace with this
                                new feature set.
              FTR=...;          Feature Tag ID [opt]
              OPT=...;          Optional Indicator [opt]
            }                   End Feature aggregate
        }                       End ACL aggregate
    }                           End REC aggregate
}%                              End *RFPK section
```

6.4.2 Feature Package Data Response (*FPKR) Section

The *FPKR section is created as a result of a request (*RFPK) in either the INQFPK contract or the MTCFPK contract. There is one response REC for each request REC in the original request message.

6.4.2.1 General Format

The following is a general format for the response message:

```
*FPKR{                          Feature Package Response section
   REC{                         Record {1-N}, where N equals the number
                                of Feature Package Responses
      CTL{                      Control data
         CTC=...;               Control code
         IERC=...;              Input error return code {0-N}, where the
                                number of input errors.
         FPRC=...;              Feature package return code {0-1}
         PROP=...;              Propagate option
         ALL=...;               Request all data option [opt]
      }                         End CTL aggregate
      ACL{                      Action line {1,2} - Original Request
         ACT=...;               Action code
         PROD=...;              Product Name
```

```
            NETYP=...;                  Switch Type
            NEGEN=...;                  Switch Generic
            FPKG=...;                   Feature Package Name
            SERVICE=...;                Type of service {0-3}
            FEATURE{                    Feature {0-N} where N equals the number
                                        of features within a package.
                FTR=...;                Feature Tag ID
                OPT=...;                Optional Indicator [opt]
            }                           End Feature aggregate
        }                               End ACL aggregate
        PRODUCT{                        Product Data {0-1}
           PROD=...;                    Product Name
           NETYP=...;                   Network Element Type
           NEGEN=...;                   Network Element Generic
           FPDATA{                      Feature Package Data {0-N}
               FPKG=...;                Feature Package Name
               SERVICE=...;             Type of service {0-3}
               FEATURE{                 Feature {0-N} where N equals the number
                                        of feature within a package.
                   FTR=...;             Feature Tag ID
                   OPT=...;             Optional Indicator
                   DESC{                Feature Descriptions {1-3}
                       SERVICE=...;     Service type <R, B, C>
                       FTRDES=...;      Description Long Name
                   }                    End of DESC aggregate
               }                        End FEATURE aggregate
           }                            End FPDATA aggregate
         }                              End PRODUCT aggregate
         ERROR{                         Error aggregate {0-N}
              FTR=...;                  Feature Tag ID
              SERVICE=...;              Service Type
               IERC=...;                Input Error Code
         }                              End Error aggregate
      }                                 End REC aggregate
}%                                      End *FPKR section
```

The response message contains the ACL from the original request message, plus additional data (FPDATA aggregate from the inquiry) and return codes. Two return codes will be present in the response section. They are the IERC (Input Error Return Code) and the FPRC (Feature Package Return Code). The IERC may contain multiple error codes indicating input errors or incorrect formats in the FCIF message. The FPRC will return the result of processing the request. The ERROR aggregate contains the IERC code to indicate which FTR or SERVICE the error belongs to.

See Appendix D for a listing of the IERC and FPRC codes.

6.5 Switch-Related Data Maintenance Contracts

This section describes two contracts, INQNED and MTCNED, which are used to maintain switch-related data. Switch-related data encompasses network element, switch parameter, and switch feature exception data. The INQNED contract supports the inquiry† of switch-related data, while the MTCNED contract supports the maintenance (build, change, and remove) of switch-related data. There is no maintenance associated with the switch parameter table, since it consists of Bellcore-provided data.

Both contracts consist of a request and a response message and share the *RNED, *NEDR, *RXCP, and *XCPR sections listed below.

- The FCIF sections in the request message are as follows:

*C1	Control Header section
*PLHDR	Provisioning Platform Header section
*SNSHDR	SNS Header section (optional)
*RNED	Request for Network Element Data section
*RXCP	Request for Switch Exception Data section

- The FCIF sections in the response message are as follows:

*C2	Control Header section
*PLHDR	Provisioning Platform Header section
*UMSG	User Message section (optional)
*SNSHDR	SNS Header section
*NEDR	Network Element Data Response section
*XCPR	Switch Exception Data Response section

See Section 3 in this document for more information on the *C1, *C2, *PLHDR, *UMSG, and *SNSHDR FCIF sections.

6.5.1 General Contract Request Information

The contracts are designed so that multiple requests may be initiated in one contract invocation (request message). This is denoted by the REC aggregate that spans from {1-N}. Therefore, the INQNED contract may contain multiple inquiry requests, while the MTCNED contract may contain multiple requests for

† The inquiries in maintenance are designed for the support of the maintenance functionality.

builds, changes, and deletes. These requests are structured in terms of the following FCIF sections.

The *RNED section supports the maintenance of network element and switch parameter data:

- Inquiry of Network Element and Switch Parameter Data (INQNED)
- Build Network Element Data only (MTCNED)
- Modify Network Element Data only (MTCNED)
- Remove Network Element Data only (MTCNED).

The *RXCP section supports the maintenance of Switch Feature Exception Data:

- Inquiry of Switch Feature Exception Data (INQNED)
- Build Switch Feature Exception Data (MTCNED)
- Modify Switch Feature Exception Data (MTCNED)
- Remove Switch Feature Exception Data (MTCNED).

6.5.2 Request Network Element Data (*RNED) Section

The *RNED section contains the data needed to perform inquiries and maintenance to network element and switch parameter data.

6.5.2.1 General *RNED Section Format

The data items† for the *RNED section are:

*RNED{	Request for Network Element Data
REC{	Record {1-N}, where N equals the number of Network Element Inquiry Requests
CTL{	Control data

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
            CTC=...;        Control code:
                            <I=Inquiry,
                              B=Build,
                              C=Change,
                              R=Remove>
        }                   End CTL aggregate
        ACL{                Action line {1,2}
            ACT=...;        Action code:
                            <O=Old,
                              N=New,
                              R=Replace>
            CLLI=...;       Network element CLLI code
            NETYP=...;      Network element type
            NEGEN=...;      Network element generic
            WC=...;         Wire center
        }                   End ACL aggregate
    }                       End REC aggregate
}%                          End *RNED section
```

The *RNED section contains a CTL aggregate and one or more ACL aggregates. The CTL aggregate identifies certain control information. In particular, the control code (CTC) is used to determine the type of control action to be performed:

- Inquire about network element data (CTC = I)
- Add/Build another network element ID (CTC = B)
- Change network element data (CTC = C)
- Remove network element data (CTC = R).

The ACL aggregate contains the specific information required to perform the switch-related data inquiry or update. The information required in each ACL aggregate varies, depending upon the CTC that appears in the CTL aggregate.

The following subsections explain how the *RNED section should be constructed, depending upon the CTC value in the CTL aggregate.

6.5.2.2 *RNED Section Format: Network Element Inquiry (INQNED)

An inquiry for network element data requires a CLLI code for network element ID.

The following data items may appear in the *RNED section:

```
*RNED{                      Request for Network Element Data
    REC{                    Record {1-N}, where N equals the number of
                            Network Element Data Requests
        CTL{                Control data
            CTC=I;          Control Code = Inquiry
        }                   End CTL aggregate
        ACL{                Action line
            CLLI=...;       Network element ID (CLLI code)
        }                   End ACL aggregate
    }                       End REC aggregate
}%                          End *RNED section
```

6.5.2.3 *RNED Section Format: Build Network Element (MTCNED)

To add a network element ID with related information requires a CTC of "B" and an ACT of "N". The required tags in the ACL aggregate are CLLI, NETYP, NEGEN, and WC.

The following data items may appear in the *RNED section:

```
*RNED{                      Request for Network Element Data
    REC{                    Record {1-N}, where N equals the number of
                            Network Element Data Requests
        CTL{                Control data
            CTC=B;          Control code - Build
        }                   End CTL aggregate
        ACL{                Action line
            ACT=N;          Activity - new
            CLLI=...;       Network element CLLI code
            NETYP=...;      Network element type
            NEGEN=...;      Network element generic
            WC=...;         Wire center
        }                   End ACL aggregate
    }                       End REC aggregate
}%                          End *RNED section
```

6.5.2.4 *RNED Section Format: Remove Network Element (MTCNED)

To delete a network element ID and its related data, a CTC of "R" and an ACT of "O" are required in the CTL aggregate. In the ACL aggregate, the CLLI code for the network element ID is required.

The following data items may appear in the *RNED section:

```
*RNED{                          Request for Network Element Data
    REC{                        Record {1-N}, where N equals the number of
                                Network Element Data Requests
        CTL{                    Control data
            CTC=R;              Control code - Remove data
        }                       End CTL aggregate
        ACL{                    Action line
            ACT=O;              Activity is against old data
            CLLI=...;           Network element CLLI code
        }                       End ACL aggregate
    }                           End REC aggregate
}%                              End *RNED section
```

6.5.2.5 *RNED Section Format: Change Network Element (MTCNED)

There are two types of changes (CTC = C) allowed for network element data. The first provides an old and new ACL with action codes (ACT) of "O" and "N". This allows updates for key fields. The old ACL holds old key values and old tag values, reflecting the current data image, while the new ACL holds the new values that allow the old data to be changed to the new data. *The lack of a tag in the old ACL implies that the value represents its default value (0 for integer, or space for character). The lack of a tag in the new ACL implies that the value of the tag will be set to its default value.* Comparing the values in the old ACL aggregate with the current database image ensures proper sequencing of the FCIF messages.

The second type of change contains one ACL with an action code of "R" for replace. The old keys are given with only the tags that contain the new values. To remove an existing value without assigning a new value, it must be explicitly assigned, dependent upon its data type. The assignment for integers is "TAG=0;". Characters are assigned "TAG= ;". This type of change is flexible since it provides the ability to change fields without prior knowledge of their old values. This method does not allow changes to key fields.

6.5.2.5.1 *RNED Section Format: Change Network Element (MTCNED) ACT=O/N

The following data items may appear in the *RNED section:

```
*RNED{                        Request for Network Element Data
    REC{                      Record {1-N}, where N equals the number of
                              Network Element Data Requests
        CTL{                  Control data
            CTC=C;            Control code - Change
        }                     End CTL aggregate
        ACL{                  Action line
            ACT=O;            Activity - old
            CLLI=...;         Network element CLLI code
            NETYP=...;        Network element type
            NEGEN=...;        Network element generic
            WC=...;           Wire center
        }                     End ACL aggregate
        ACL{                  Action line
            ACT=N;            Activity - data
            CLLI=...;         Network element CLLI code
            NETYP=...;        Network element type
            NEGEN=...;        Network element generic
            WC=...;           Wire center
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RNED section
```

6.5.2.5.2 *RNED Section Format: Change Network Element (MTCNED) ACT=RPL

In this type of change, a CTC of "C" and only one ACL aggregate is required. As a reminder, you cannot change the key field with this type of change.

The following data items may appear in the *RNED section:

```
*RNED{                        Request for Network Element Data
    REC{                      Record {1-N}, where N equals the number of
                              Network Element Data Requests
        CTL{                  Control data
            CTC=C;            Control code - Change data
        }                     End CTL aggregate
        ACL{                  Action line
            ACT=R;            Activity - replace
            CLLI=...;         Network element CLLI code (old key)
            NETYP=...;        Network element type    [opt]
            NEGEN=...;        Network element generic [opt]
            WC=...;           Wire center  [opt]
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RNED section
```

6.5.3 Request Switch Exception Data (*RXCP) Section

The *RXCP section contains the data needed to perform inquiries and maintenance to switch feature exception data.

6.5.3.1 General *RXCP Section Format

The data items† for the *RXCP section are:

```
*RXCP{                        Request for Switch Feature Exception Data
    REC{                      Record {1-N}, where N equals the number of
                              Switch Feature Exception Requests
        CTL{                  Control data
            CTC=...;          Control code:
                              <I=Inquiry,
                                B=Build,
                                C=Change,
                                R=Remove>
        }                     End CTL aggregate
        ACL{                  Action line {1,2}
```

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
                 ACT=...;          Action code:
                                   <O=Old,
                                   N=New,
                                   R=Replace>
                 CLLI=...;         Network element CLLI code
                 FTR=...;          Feature tag
                 SERVICE=...;      Service Type {1-3}:
                                   B=Business,
                                   C=Centrex,
                                   R=Residence>
            }                      End ACL aggregate
       }                           End REC aggregate
}%                                 End *RXCP section
```

The *RXCP section contains a CTL aggregate and one or more ACL aggregates. The CTL aggregate identifies certain control information. In particular, the control code (CTC) is used to determine the type of control action to be performed as previously described in the *RNED section.

The ACL aggregate contains the specific information required to perform the network element inquiry or update. The information required in each ACL aggregate varies, depending upon the CTC that appears in the CTL aggregate.

The following subsections explain how the *RXCP section should be constructed, depending upon the CTC value in the CTL aggregate.

6.5.3.2 *RXCP Section Format: Switch Feature Exception Inquiry (INQNED)

An inquiry for switch feature exception data requires a CLLI code for the network element ID. In this case, all the feature exceptions at that switch are returned. If a specific feature tag is given, then only that specific feature exception is returned.

The following data items may appear in the *RXCP section:

```
*RXCP{                             Request for Switch Feature Exception Data
     REC{                          Record {1-N}, where N equals the number of
                                   Switch Feature Exception Data Requests
          CTL{                     Control data
               CTC=I;              Control Code = Inquiry
          }                        End CTL aggregate
          ACL{                     Action line
```

```
            CLLI=...;             Network element ID (CLLI code)
            FTR=...;              Feature tag [opt]
        }                         End ACL aggregate
    }                             End REC aggregate
}%                                End *RXCP section
```

6.5.3.3 *RXCP Section Format: Build a Switch Feature Exception (MTCNED)

To add feature exceptions at a specific switch requires a CTC of "B" and an ACT of "N". The switch and feature tag must also be specified.

The following data items may appear in the *RXCP section:

```
*RXCP{                            Request for Switch Feature Exception Data
    REC{                          Record {1-N}, where N equals the number of
                                  Switch Feature Exception Data Requests
        CTL{                      Control data
            CTC=B;                Control code - Build
        }                         End CTL aggregate
        ACL{                      Action line
            ACT=N;                New data
            CLLI=...;             Network element CLLI code
            FTR=...;              Network element type
            SERVICE=...;          Service type {1-3}
        }                         End ACL aggregate
    }                             End REC aggregate
}%                                End *RXCP section
```

6.5.3.4 *RXCP Section Format: Remove a Switch Feature Exception (MTCNED)

To remove all the feature exceptions at a particular switch only the switch ID (CLLI) is required. To remove a specific feature exception at a switch, the switch ID (CLLI) and the feature tag name (FTR) are required. Either delete capability requires a CTC of "R" in the CTL aggregate, and an ACT of "O" in the ACL aggregate.

The following data items may appear in the *RXCP section:

```
*RXCP{                        Request for Switch Feature Exception Data
    REC{                      Record {1-N}, where N equals the number of
                              Switch Feature Exception Data Requests
        CTL{                  Control data
            CTC=R;            Control code - Remove data
        }                     End CTL aggregate
        ACL{                  Action line
            ACT=O;            Activity is against old data
            CLLI=...;         Network element CLLI code
            FTR=...;          Feature tag     [opt]
        }                     End ACL aggregate
    }                         End REC aggregate
}%                            End *RXCP section
```

6.5.3.5 *RXCP Section Format: Change Switch Feature Exception Data (MTCNED)

There are two types of changes allowed for Switch Feature Exception Data. Refer to the *RNED section for more information on these change types.

6.5.3.5.1 *RXCP Section Format: Change Switch Feature Exception Data (MTCNED) ACT=O/N

This type of change requires a CTC of "C" and two ACL aggregates. It supports key changes, and ensures proper sequencing.

The following data items may appear in the *RXCP section:

```
*RXCP{                        Request for Switch Feature Exception Data
    REC{                      Record {1-N}, where N equals the number of
                              Switch Feature Exception Data Requests
        CTL{                  Control data
            CTC=C;            Control code - Change
        }                     End CTL aggregate
        ACL{                  Action line
            ACT=O;            Action Code - old
            CLLI=...;         Network element CLLI code
            FTR=...;          Feature tag
            SERVICE=...;      Service type {1-3}
            }                       End ACL aggregate
            ACL{                    Action line
                ACT=N;              Activity is new data
                CLLI=...;           Network element CLLI code
                FTR=...;            Feature tag
                SERVICE=...;        Service type {1-3}
            }                       End ACL aggregate
        }                           End REC aggregate
}%                                  End *RXCP section
```

6.5.3.5.2 *RXCP Section Format: Change Switch Feature Exception Data (MTCNED) ACT=RPL

In this type of change, a CTC of "C" and only one ACL aggregate are required. The switch ID and feature tag must be specified in order to use this change function. As a reminder, you cannot change the key field with this type of change.

The following data items may appear in the *RXCP section:

```
*RXCP{                              Request for Switch Feature Exception Data
        REC{                        Record {1-N}, where N equals the number of
                                    Switch Feature Exception Data Requests
            CTL{                    Control data
                CTC=C;              Control code - Change data
            }                       End CTL aggregate
            ACL{                    Action line
                ACT=R;              Action code - replace
                CLLI=...;           Network element CLLI code
                FTR=...;            Feature tag
                SERVICE=...;        Service type {1-3}
            }                       End ACL aggregate
        }                           End REC aggregate
}%                                  End *RXCP section
```

6.5.4 Network Element Data Response (*NEDR) Section

The *NEDR section is created as a result of a request in the *RNED section of the INQNED or MTCNED contract. There is one response REC for each REC in the original request message. The section contains responses to network-related information. The *NEDR section also identifies errors and discrepancies.

6.5.4.1 General *NEDR Section Format

The data items† for the *NEDR section are shown below in a general format. The *RNED section of the request message is provided in the CTL and ACL aggregates of the corresponding *NEDR section of the response message for each REC aggregate. The items shown in regular typeface reflect the data provided in the original request (*RNED section). The items shown in bold typeface represent the response data.

*NEDR{	Network Element Data Response
REC{	Record {1-N}, where N equals the number of Network Element Responses
CTL{	Control data
CTC=...;	Control code
IERC=...;	**Input error return code {0-N}, where N=the number of input errors**
NDRC=...;	**Network data return code {0-N}**
}	End CTL aggregate
ACL{	Action line {1,2}
CLLI=...;	Network element ID (CLLI code)
NETYP=...;	Network element type
NEGEN=...;	Network element generic
WC=...;	Wire center
}	End ACL aggregate
NEDATA{	**Network element information {0-1}**
CLLI=...;	**Network element CLLI code**
NETYP=...;	**Network element type**
NEGEN=...;	**Network element generic**
WC=...;	**Wire center**
MXSCH=...;	**Maximum members in a SCH group**
MXSCHP=...;	**Maximum SCH groups to participate**
}	**End NEDATA aggregate**
XCPDATA{	**Switch feature exception data {0-N}**
FTR=...;	**Feature tag**

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
            SVC{                  SVC aggregate {1-3}
                SERVICE=...;      Service type
                FTRDES=...;       Description
            }                     End SVC aggregate
        }                         End XCPDATA aggregate
    }                             End REC aggregate
}%                                End *NEDR section
```

The first group of response data (IERC and NDRC) includes the CTL aggregate return codes. This data is returned as a response in all cases (CTC = I, B, C, R).

The NEDATA aggregate shows the additional data that may appear in the *NEDR section. This aggregate is returned upon a successful inquiry request.

The XCPDATA aggregate is returned when a change of type or generic leads to a failure in revalidating the feature exceptions against the catalog for the updated type and/or generic. This is an internal database integrity check comparing the switch feature exception data against the catalog data. The update does not take place until the failed feature exceptions are either deleted or reset. The contract returns the switch feature exceptions that failed the validation. The return codes and the additional aggregates are explained in the subsections below.

6.5.4.2 CTL Aggregate Return Codes

There are two return code tags that may be included in the CTL aggregate. In the general format above, they are shown in bold typeface within the CTL aggregate. These return code tags are:

- Input Error Return Code (IERC)

  This return code indicates an incorrect format in the FCIF message or the associated tags. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors. See Appendix D for specific IERC values and descriptions.

- Network Data Return Code (NDRC)

  The NDRC tag indicates the result of inquiries, additions, changes, replacements, and deletions of network element data. See Appendix D for specific NDRC values and descriptions.

6.5.4.3 *NEDR Section Format: Network Element Data Inquiry Response

Upon successful completion of an inquiry, the data items in the NEDATA aggregate of the *NEDR general section are returned. If unsuccessful, only the IERC and NDRC tags are returned.†

6.5.4.4 *NEDR Section Format: MTCNED Contract Response

The response to any maintenance activity (CTC = B, C, R) results in either successful or unsuccessful return codes (IERC/NDRC) found in the CTL aggregate of the response section. Other data that may possibly be returned in this case is the XCPDATA aggregate, which only appears when a type and/or generic have been changed and the exceptions fail revalidation against the catalog. The NEDATA aggregate is returned only upon a successful inquiry.

```
*NEDR{                       Network Element Data Response
    REC                      Record {1-N}, where N equals the number of
                             Network Element Responses
     CTL{                    Control data
          CTC=...;           Control code <B,C,R>
          IERC=...;          Input error return code {0-N}, where N= the
                             number of input errors
          NDRC=...;          Network data return code {0-1}
       }                     End CTL aggregate
       ACL{                  Action line {1,2} from original request
       }                     End ACL aggregate
       XCPDATA{              Exception data {1-N}, where N = the number of
                             feature exception validation failures for a switch
                             type/generic change
       }                     End XCPDATA aggregate
    }                        End REC aggregate
}%                           End *NEDR section
```

† In the case of unsuccessful changes of switch type and/or switch generic, revalidation failures of feature exceptions cause the XCPDATA aggregate to be returned.

6.5.5 Switch Feature Exception Data Response (*XCPR) Section

The *XCPR section is created as a result of a request in the *RXCP section of the INQNED or MTCNED contract. There is one response REC for each REC in the original request message. The section contains responses to switch feature exception-related information. The *XCPR section also identifies errors and discrepancies.

6.5.5.1 General *XCPR Section Format

The data items† for the *XCPR section are shown below in a general format. The *RXCP section of the request message is provided in the CTL and ACL aggregates of the corresponding *XCPR section of the response message for each REC aggregate. The items shown in regular typeface reflect the data provided in the original request (*RXCP section). The items shown in bold typeface represent the response data.

*XCPR{	Network Element Data Response
REC{	Record {1-N}, where N equals the number of Network Element Responses
CTL{	Control data
CTC = ...;	Control code
IERC = ...;	**Input error return code {0-N}, where N = the number of input errors**
FERC = ...;	**Feature exception return code {0-1}**
}	End CTL aggregate
ACL{	Action line {1,2}

ACT = ...;	Action code:
	< O = Old,
	N = New,
	R = Replace >

CLLI = ...;	Network element CLLI code
FTR = ...;	Feature tag

† Data items shown in the *general* section format do not indicate whether they are optional, conditional, or required. This information is supplied in the subsequent sections.

```
                SERVICE=...;          Service Type {1-3}:
                                         B=Business,
                                         C=Centrex,
                                         R=Residence>
        }                             End ACL aggregate
        XCPDATA{                      Exception data information {0-N}, where N=the
                                      number of feature exceptions
            FTR=...;                  Feature tag
            SVC{                      SVC aggregate {1-3}
                SERVICE=...;          Service type
                FTRDES=...;           Description
            }                         End SVC aggregate
        }                             End XCPDATA aggregate
        ERROR{                        ERROR aggregate {0-N}
                SERVICE=...;          Service type
                SVCERR=...;           Service error code
          }                           End ERROR aggregate
    }                                 End REC aggregate
}%                                    End *XCPR section
```

The first group of response data (IERC and FERC) includes the CTL aggregate return codes. This data is returned as a response in all cases (CTC = I, B, C, R). The second group of response data (the XCPDATA aggregate) shows the additional data that may appear in the *XCPR section. This data is returned only upon a successful inquiry request.

If an error is detected with the input SERVICE data, the ERROR aggregate will be generated on output. This is done to indicate which specific occurrence of SERVICE is in error, since several could be present on input.

6.5.5.2 CTL Aggregate Return Codes

There are two return code tags that may be included in the CTL aggregate. In the general format above, they are shown in bold typeface within the CTL aggregate. These return code tags are:

- Input Error Return Code (IERC)

  This return code indicates an incorrect format in the FCIF message or the associated tags. The value associated with the tag indicates the error condition. This tag may appear multiple times to identify multiple input errors. See Appendix D for specific IERC values and descriptions.

- Feature Exception Return Code - (FERC)

  The FERC tag indicates the result of inquiries, additions, changes, replacements, and deletions of feature exception data. See Appendix D for specific FERC values and descriptions.

6.5.5.3 *XCPR Section Format: Switch Feature Exception Data Inquiry Response

Upon successful completion of an inquiry, the data items in the XCPDATA aggregate of the *XCPR general section are returned. If unsuccessful, only the IERC and FERC tags are returned.

6.5.5.4 *XCPR Section Format: MTCNED Contract Response

The response to any maintenance activity (CTC = B, C, R) results in either a successful or unsuccessful return code. Data is not returned as it is in the case of an inquiry request. Therefore, for this case, the XCPDATA aggregate is never returned. Only the IERC and NDRC tags are returned. These tags hold the input errors and return codes described previously.

```
*XCPR{                     Network Element Data Response
    REC                    Record {1-N}, where N equals the number of
                           Network Element Responses
     CTL{                  Control data
          CTC=...;         Control code <B, C, R>
          IERC=...;        Input error return code {0-N}, where N=the
                           number of input errors
          FERC=...;        Feature exception return code {0-1}
      }                    End CTL aggregate
      ACL{                 Action line {1,2} from original request
      }                    End ACL aggregate
      ERROR{               ERROR aggregate {0-N}
         SERVICE=...;      Service type
         SVCERR=...;       Service error code
      }                    End ERROR aggregate
    }                      End REC aggregate
}%                         End *XCPR section
```

If an error is detected with the input SERVICE data, the ERROR aggregate will be generated on output. This is done to indicate which specific occurrence of SERVICE is in error, since several could be present on input.