**Table 6–5.** Required Tables for the SOAC and CRS Interfaces

| The NSDB tables | SOAC-NSDB-WFA/C * INTERFACE | CRS-NSDB-WFA/C ** INTERFACE |
|---|---|---|
| ODB IMAGE | required | |
| ODB AUDIT | required | |
| ODB WCLLI – WCTR | required | |
| ODB USOC OPTIONS | optional | |
| ODB SEC TO SRC | required | |
| ODB CO OPTIONS | required | required |
| ODB EXCEPTIONS | optional | optional |
| ODB DATA ROUTING | required | required |
| **TCM tables and screens** | | |
| TCM USER CONTROL | required | required |
| TCM Network Administration screen VJMPNET | 17 required | 3 required |
| TCM Network Administration screen OMPNET | 7 required *** | 1 required *** |

* The NMA system (for ISDN monitoring) can also receive ISDN involved Service Orders and Work Orders from SOAC. The NSDB and TCM tables listed for the SOAC-NSDB-WFA/C interface provide SOAC data to the NMA system.

** The NMA system also receives TN swap data from CRS. The NSDB and TCM tables listed for the CRS-NSDB-WFA/C interface provide TN swap updates to the NMA system.

*** These screens are created and maintained by the WFA/C System Administrator. They are listed here for reference only.

# 6.2 CRS-NSDB-WFA/C Interface

## 6.2.1 Description

*Purpose*

The purpose of this interface is to manage data that is changed by customers using a Centrex Rearrangement System (CRS) product for telephone number swaps.

*Data Source*

The source of data (really changes of data) is a CRS system. CRS is a generic term for products that enable customers to make limited changes in their service without service orders. The CRS change activity that is supported by NSDB is called "Telephone Number swap" (TN swap). TN swap enables customers to swap the telephone number of one of their locations with that of another. If the change occurs on a telephone number that is stored by NSDB, then the Line Record database must be updated. In addition, WFA/C must be notified. This interface is used to manage changes of this sort and to notify WFA/C.

The CRS-NSDB-WFA/C interface can be used only if the interface requirements of NSDB are met.

*TCM Paths*

Three TCM paths are used with this interface. The path name SWAP is used for the change data that comes from CRS to NSDB. A second SWAP pathname is used for transmitting change data between NSDB and WFA/C. A third path name is ODBTCMAK; it is used for internal acknowledgements to NSDB.

## 6.2.2 Setting Up the Interface

The CRS-NSDB-WFA/C interface can be used only if the interface requirements of NSDB are met.

The NSDB System Administrator must create and maintain the 3 NSDB TCM Network Administration screens for this interface. The ODB DATA ROUTING table is used to distribute the CRS change data to the appropriate WFA/C centers.

# 6.3 NSDB-ITS Automated Platform (FCIF) Interface

## 6.3.1 Description

*Purpose*

The purpose of this interface is to provide to the Integrated Test System (ITS) all of the circuit data necessary for testing designed and nondesigned analog and digital services.

*Data Source*

The sources of this circuit data are the TIRKS system and the Service Order Analysis and Control (SOAC) system. NSDB receives special circuit, message, and carrier circuit data from the TIRKS Work Order Record Details (WORD) document. NSDB receives service orders and work orders from SOAC. When there is a conflict between TIRKS and SOAC loop data for special service circuits, NSDB will use the most recent loop information, which is determined via the match/merge process. ITS requests circuit data from NSDB when it needs to perform a circuit test.

## 6.3.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. The list below displays the required and optional tables and screens necessary to support the NSDB-ITS Automated Platform interface. For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired.

The interface between NSDB, TCM, and ITS is normally set up through a cooperative effort between the NSDB and ITS system administrators and communications administrators.

- TCM screens and tables (for the NSDB side):

  TCM Network Administration screens, VJMPNET
  (3 required: send, receive, acknowledgement)

  TCM USER CONTROL table

- The NSDB tables:

  ODB CO OPTIONS
  ODB CTVAL
  ODB EXCEPTIONS
  ODB HECIG TO EQP
  ODB MCO OPTIONS

ODB OFFICE DATA
ODB SCVAL
ODB SEC TO SRC
ODB SERVICE CODE

### 6.3.3 Maintaining the Interface

*Coordination*

Maintaining this interface requires a coordinated effort among personnel from NSDB and ITS. Problems with analog and digital circuit data are detected by using the online validation process or the BMP run, VJODU01. (Refer to chapter 8 for details about this run, and Appendix A for a description of possible error messages.)

*Table Maintenance*

The flow of data and any error messages for this interface are controlled by the NSDB and TCM tables and screens listed above. Changes in the business that influence the data flow or error processing methods could require tuning of one or more tables. (Refer to chapter 7 for NSDB table descriptions.)

The distribution of error messages is controlled by the ODB EXCEPTIONS, ODB MCO OPTIONS, and ODB CO OPTIONS tables. When error messages are desired at a new or different location, these tables should be reviewed. The ODB EXCEPTIONS table allows for the suppression of messages. This table should be periodically reviewed to ensure that needed messages are not being suppressed.

For digital services, special attention should be paid to the ODB HECIG TO EQP table. This table requires tuning when changes are made to the HECIG information received from TIRKS. If this is not done, ITS users will not get the information they require to test the circuits properly. The ODB OFFICE DATA table is also tied to the ODB HECIG TO EQP table. The ODB OFFICE DATA table is used to validate the Point of Termination (POT) locations used for DDS circuits.

The ODB SCVAL and ODB CTVAL are used together to select the types of digital circuits that are subject to online validation. Circuit Type and Service Class are used as keys to select circuits for validation.

*Maintenance Runs*

The Bulk Edit For NSDB Databases, VJODU01, is used to validate designs from the TIRKS system to ensure that ITS can use them. The validation of DDS, DS1, and CLFI type circuits can also take place using online validation.

*Useful Formats*

The NSDB VJDBCD format can be used to apply the desired Service Rate when multiple rates apply to a given service code.

## 6.4 TIRKS-NSDB-ITS/Digital (DSECT) Interface

### 6.4.1 Description

*Purpose*

The purpose of this interface is to provide the Integrated Test System (ITS) with.circuit data necessary for testing designed digital services.

*Data Source*

The source of this circuit data is the TIRKS system. NSDB receives the circuit details from the Work Order Record Details (WORD) document. This 'event' is known to the TIRKS system as ISSUWD. The WORD document is the basis for the circuit record kept by NSDB for use by ITS. ITS requests circuit data from NSDB when it needs to perform a circuit test.

### 6.4.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. The list below displays the required and optional tables and screens necessary to support the TIRKS-NSDB-ITS (DSECT) interface. For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired.

The NSDB, TIRKS system, and ITS tables are normally set up through a cooperative effort between the NSDB, TIRKS, and ITS system administrators.

- The TIRKS system tables:

  EDIIS EVENT FUNC
  EDIIS NSDB OPTS
  EDIIS OPTION CD
  EDIIS UCT

- The NSDB tables:

  ODB CO OPTIONS
  ODB CTVAL
  ODB EXCEPTIONS
  ODB HECIG TO EQP
  ODB MCO OPTIONS
  ODB OFFICE DATA
  ODB SCVAL
  ODB SEC TO SRC
  ODB SERVICE CODE

### 6.4.3 Maintaining the Interface

*Coordination*

Maintaining this interface requires a coordinated effort among personnel from TIRKS, NSDB, ITS, and the Circuit Provisioning Center (CPC). Problems with digital circuit data are detected by using the on-line validation process, or the BMP run VJODU01. (Refer to chapter 8 for details about this run, and Appendix A for a description of possible error messages.)

*Table Maintenance*

The flow of data and any error messages for this interface are controlled by the TIRKS and NSDB tables shown above. Changes in the business that influence the data flow or error processing methods could require tuning of one or more tables. (Refer to chapter 7 for NSDB table descriptions.)

Special attention should by paid to the HECIG TO EQP table. This table requires tuning when changes are made to the HECIG information received from TIRKS. If this is not done, ITS users will not get the information they require to test the circuits properly. The ODB OFFICE DATA table is also tied to the HECIG TO EQP table. The ODB OFFICE DATA table is used to validate the Point of Termination (POT) locations used for DDS circuits.

The ODB SCVAL and ODB CTVAL are used together to select the type of circuits that are subject to online validation. The Circuit Type and Service Class are used as keys to select circuits for validation.

The distribution of error messages is controlled by the ODB EXCEPTIONS, ODB MCO OPTIONS, and ODB CO OPTIONS tables. When error messages are desired at a new or different location, these tables should be reviewed. The ODB EXCEPTIONS table allows for the suppression of messages. This table should be periodically reviewed to ensure that needed messages are not being suppressed.

*Maintenance Runs*

The Bulk Edit For NSDB Databases, VJODU01, is used to validate designs from the TIRKS system to ensure that ITS/Digital can use them. The validation of DDS, DS1, and CLFI type circuits can also take place using online validation.

*Useful Formats*

The NSDB format VJDBCD can be used to apply the desired Service Rate when multiple rates apply to a given service code.

## 6.5 SOAC-NSDB-WFA/C Interface

### 6.5.1 Description

*Purpose*

The purpose of this interface is to enable NSDB to receive and store order and circuit data from SOAC so that it can be made available to the user systems.

*Data Source*

The SOAC system is the source of data for this interface. SOAC data is used by NSDB to create and store line records that are viewed by WFA/C and WFA/DO through online formats. In addition to creating line records, NSDB also sends the order to WFA/C where it is used during the administration of installation activities. The SOAC data passed through this interface can be for designed circuits, nondesigned circuits, or both. Tables in the SOAC system control the type of data that is passed to NSDB.

In addition to service orders, work orders that are initiated in the LFACS system are received by NSDB. Work orders received by NSDB are not passed to WFA/C, but are used to provide redundancy management between the LFACS data and the NSDB Line Record databases.

*TCM Paths*

There are 17 TCM paths associated with this interface.

Six path names are used to send SOAC Service Order data to NSDB. (These path names are also used in the NSDB to WFA/C link. These six path names, each used twice, account for 12 of the 17 total TCM paths.) These path names are:

SOACNAS — This path is for Assignment messages from SOAC to NSDB, and from NSDB to WFA/C.

SOACNCN — This path is for Cancel messages from SOAC to NSDB, and from NSDB to WFA/C.

SOACNCP — This path is for Completion messages from SOAC to NSDB, and from NSDB to WFA/C.

SOACNDD — This path is for Order Due Date change messages from SOAC to NSDB, and from NSDB to WFA/C.

SOACNNC — This path is for correction messages that do not involve NSDB–stored data. The SOACNNC path name is used from SOAC to NSDB, and from NSDB to WFA/C.

SOACNPL — This path is for Planning messages from SOAC to NSDB, and from NSDB to WFA/C.

Two more path names are used for SOAC Work Order data that is sent to NSDB. (Work Orders are not sent on to WFA/C.) These path names are:

SOACWCF This path is for Central Office Work Order messages from SOAC to NSDB.

SOACWSP This path is for Outside Plant Work Order messages from SOAC to NSDB.

Two more paths are used to send acknowledgements. These path names are:

ODBTCMAK This path is for internal NSDB acknowledgements.

SMPACKSO This path is for positive acknowledgements from NSDB to SOAC.

The final path name is:

MATCH This path is for sending 'match' messages from NSDB to WFA/C. (This message tells WFA/C if matching TIRKS design data with SOAC order data was successful or not.)

### 6.5.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. The list on the following page displays the required and optional tables and screens necessary to support the SOAC-NSDB-WFA/C interface.

The SOAC system, TCM, and WFA/C tables are normally set up through a cooperative effort between the NSDB, SOAC, and WFA/C system administrators and communications administrators. For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired.

- TCM screens and tables (for the NSDB side):

  TCM USER CONTROL

  TCM Network Administration screens, VJMPNET
  At least 17 required:
  receive from SOAC (8 required)
  internal acknowledgement (1 required)
  acknowledgement back to SOAC (1 required)
  * send to WFA/C (7 required)

- TCM screens and tables (for the WFA/C side **):

  TCM USER CONTROL
  TCM Network Administration screen, OMPNET
  receive from NSDB (7 required)

---

* Each WFA/C system receiving data from NSDB requires its own set of receive TCM Network Administration screens for this interface.

** These TCM tables and screens are set up by the WFA/C System Administrator. They are listed here for reference only.

- NSDB tables:

  ODB AUDIT
  ODB CO OPTIONS
  ODB USOC OPTIONS
  ODB IMAGE
  ODB WCLLI – WCTR
  ODB DATA ROUTING

### 6.5.3 Maintaining the Interface

Ongoing coordination between the NSDB, SOAC, and WFA/C system administrators is required to maintain this interface. Each system has tunable tables that can influence the other systems, and the resultant overall performance of the interface.

## 6.6 TIRKS-NSDB-NMA Facilities Interface

### 6.6.1 Description

*Purpose*

The purpose of this interface is to enable NSDB to store and send TIRKS system pending carrier circuit data to the NMA system. Upon receiving this pending carrier facility data, the NMA system performs pre-service monitoring to ensure that the service is ready for turn-up to the customer. The carrier alarm features are tested at this time.

*Data Source*

The TIRKS system provides the data for this interface. This 'event' is known to the TIRKS system as ISSUWD. NSDB passes this circuit data to the NMA system on the pre–service notification date received from WFA/C or the TIRKS system.

*TCM Paths*

This interface uses the same TCM paths that are used by the TIRKS-NSDB-NMA/Switch interface. NSDB receives the TIRKS data on the TCM path called EDIISODB with a scenario type of R for receive. This data is acknowledged internally on a path called ODBTCMAK. Once acknowledged, the data is passed to the NMA system on the path called NDBTONMA. The NMA system acknowledges the messages from NSDB on the path NMATCMAK.

### 6.6.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. These tables are the same ones used in the TIRKS-NSDB-NMA Switch interface described later. If the TIRKS-NSDB-NMA Switch interface is activated, the TIRKS-NSDB-NMA Facilities interface is also activated.

The list below displays the required and optional tables and screens necessary for NSDB to receive the TIRKS data and send it to the NMA system.

There are additional TIRKS system tables and WFA/C tables that must be set up by the WFA/C System Administrator to ensure that the NMA system receives the data when it becomes in-effect, when it is updated, or when it is disconnected. These additional tables are listed in the *WFA/C System Administrator's Guide*, BR 190-513-092.

For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired. The TIRKS system, TCM, and NMA tables are normally set up through a cooperative effort between the NSDB, TIRKS, and NMA system administrators and communications administrators.

- The TIRKS system tables:

  EDIIS EVENT FUNC
  EDIIS UCT
  EDIIS OPTION CD

- TCM screens and tables (for the TIRKS side):

  TCM Network Administration screen, RMPNET
  TCM USER CONTROL table

- TCM screens and tables (for the NSDB side):

  TCM Network Administration screens, VJMPNET
  At least 4 are required:
  receive from the TIRKS system,
  internal NSDB acknowledgement for the TIRKS data,
  * send to the NMA system,
  * external acknowledgement from the NMA system

  TCM USER CONTROL table

- NSDB tables:

  ODB ROUTEBY
  ODB DATA ROUTING
  ODB CO OPTIONS
  ODB MCO OPTIONS
  ODB SEC TO SRC
  ODB EXCEPTIONS (recommended optional table)

### 6.6.3 Maintaining the Interface

The routing of data to the NMA system requires the use of the ODB ROUTEBY and ODB DATA ROUTING tables. If the routing method changes or new NMA machines are added, these tables must be updated.

* Each NMA system served requires its own set of send and acknowledgement TCM Network Administration screens.

## 6.7 TIRKS-NSDB-NMA Pass–Through Interface

### 6.7.1 Description

*Purpose*

The purpose of this interface is to provide the inventory and assignment data required for network monitoring to the NMA system.

*Data Source*

The source of the circuit data is the TIRKS system. NSDB passes this data through to the NMA system upon updates to in-effect facilities and equipment. NSDB does not store the data passed in this interface. The Pass Through 'event' is known to the TIRKS system as 'NMADATA', and is selected on the TIRKS table called EDIIS EVENT FUNC.

*TCM Paths*

NSDB receives the TIRKS data on the TCM path called EDIISODN with a scenario type of R for receive. The data is passed to the NMA system on the path called ODBTONMA. The NMA system acknowledges the messages from NSDB on the path NMATCMAK.

### 6.7.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. The list below displays the required and optional tables and screens necessary to support the TIRKS-NSDB-NMA Inventory interface. For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired.

The TIRKS system, NMA system, and TCM tables are normally set up through a cooperative effort between the NSDB, TIRKS, and NMA system administrators and communications administrators.

- The TIRKS system tables:

  EDIIS EVENT FUNC
  EDIIS OPTION CD
  EDIIS UCT

- TCM screens and tables (for the TIRKS side):

  TCM Network Administration screen, RMPNET
  TCM USER CONTROL table

- TCM screens and tables (for the NSDB side):

  TCM Network Administration screens, VJMPNET
  At least 4 are required:
  receive from the TIRKS system,
  internal NSDB acknowledgement for the TIRKS data,
  * send to the NMA system,
  * external acknowledgement from the NMA system

- TCM USER CONTROL table
- NSDB tables:

  ODB CO OPTIONS
  ODB MCO OPTIONS
  ODB EXCEPTIONS
  ODB DATA ROUTING
  ODB ROUTEBY
  ODB DATA ROUTING

### 6.7.3 Maintaining the Interface

The routing of data to the NMA system requires the use of the ODB ROUTEBY and ODB DATA ROUTING tables. If the routing method changes or new NMA machines are added, these tables must be updated.

* Each NMA system served requires its own set of send and acknowledgement TCM Network Administration screens.

## 6.8 TIRKS-NSDB-NMA Switch Interface

### 6.8.1 Description

*Purpose*

The purpose of this interface is to provide the data required for the monitoring of message trunks, switched special services circuits, and Basic Rate ISDN by the NMA system.

*Data Source*

One source of the NMA/Switch data is the EDIIS module of the TIRKS system. The EDIIS 'event' used here is called ISSUWD. NSDB passes ISSUWD data to the NMA system upon due date completion of an order by WFA/C or the TIRKS system. Subsequent updates, disconnects, or reissues are sent to the NMA system.

Another data source is SOAC. NMA/Switch needs SOAC data to monitor ISDN circuits. SOAC Work Orders and Service Orders are sent to NMA/Switch upon completion. For more information about the designed and nondesigned data that is passed to the NMA system, refer to BR 190-534-321, the *NSDB Contract Dictionary.*

*TCM Paths*

NSDB receives the TIRKS data on the TCM path called EDIISODB with a scenario type of R for receive. This data is acknowledged internally on a path called ODBTCMAK. Once acknowledged, the data is passed to the NMA system on the path called NDBTONMA. The NMA system acknowledges the messages from NSDB on the path NMATCMAK.

NSDB also receives SOAC data. These TCM paths are described in the SOAC-NSDB-WFA/C Interface section.

### 6.8.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. The list on the following page displays the required and optional tables and screens necessary for NSDB to receive the TIRKS data and send it to the NMA system.

For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired. The TIRKS system, TCM, and NMA tables are normally set up through a cooperative effort between the NSDB, TIRKS, and NMA system administrators and communications administrators.

- The TIRKS system tables:

  EDIIS EVENT FUNC
  EDIIS UCT
  EDIIS OPTION CD

- TCM screens and tables (for the TIRKS side):

  TCM Network Administration screen, RMPNET
  TCM USER CONTROL table

- TCM screens and tables (for the NSDB side):

  TCM Network Administration screens, VJMPNET
  At least 4 are required:
  receive from the TIRKS system,
  internal NSDB acknowledgement for the TIRKS data,
  * send to the NMA system,
  * external acknowledgement from the NMA system

  TCM USER CONTROL table

- NSDB tables:

  ODB ROUTEBY
  ODB DATA ROUTING
  ODB CO OPTIONS
  ODB MCO OPTIONS
  ODB MON SWITCH
  ODB SERVICE CODE
  ODB SEC TO SRC
  ODB EXCEPTIONS (recommended optional table)

### 6.8.3 Maintaining the Interface

The routing of data to the NMA system requires the use of the ODB ROUTEBY and ODB DATA ROUTING tables. If the routing method changes or new NMA machines are added, these tables must be updated.

---

* Each NMA system served requires its own set of send and acknowledgement TCM Network Administration screens.

## 6.9 TIRKS–NSDB & WFA/C–OPS/INE Interface

### 6.9.1 Description

*Purpose*

The purpose of this interface is to provide the circuit data necessary for OPS/INE (Operations System/Intelligent Network Elements) to remotely provision the special circuits and message/carrier circuits riding through or terminating on intelligent network elements. Examples of network elements are Digital Cross–connect Systems (DCS), Electronic Digital Signal Cross–connects (EDSX), SONET Add–Drop Multiplexers (ADMs), and Automated Digital Terminal Systems (ADTS).

*Data Source*

The source of this circuit data is the TIRKS system. NSDB receives circuit details when the Work Order Record Details (WORD) document is issued. This "event" is known to the TIRKS system as ISSUWD. The WORD document is the basis for the circuit record kept by NSDB for use by OPS/INE.

*TCM Paths*

NSDB receives the TIRKS data on the TCM path called EDIISODB with a scenario type of R for receive. The message is acknowledged internally on the ODBTCMAK path with a scenario type of I for internal. The data is sent to OPS/INE automatically via a TCM path.

### 6.9.2 Setting Up the Interface

The NSDB system administrator must create and maintain several tables and screens to support this interface. The list below displays the required and optional tables and screens necessary to support the TIRKS-NSDB & WFA/C–OPS/INE interface. For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired.

The TIRKS, TCM, and OPS/INE tables are normally set up through a cooperative effort between the NSDB, TIRKS, and OPS/INE system administrators and communications administrators.

- The TIRKS system tables:

  EDIIS EVENT FUNC
  EDIIS UCT
  EDIIS OPTION CD
  EDIIS OPCDB OPTS

- TCM screens and tables (for the TIRKS side):

  TCM Network Administration screen, RMPNET
  TCM USER CONTROL table

- TCM screens and tables (for the NSDB side):

  TCM Network Administration screens, VJMPNET
  3 required: 1–send to OPS/INE
  1–acknowledgement to TIRKS
  1–acknowledgement from OPS/INE

  TCM USER CONTROL table

- The NSDB tables:

  ODB BANKCODES
  ODB CO OPTIONS
  ODB DATA ROUTING
  ODB DATAPORT
  ODB EXCEPTIONS
  ODB HECIG TO EQP
  ODB LOCATIONS
  ODB MCO OPTIONS
  ODB ROUTEBY
  ODB SEC TO SRC

### 6.9.3 Maintaining the Interface

*Coordination*

Maintaining this interface requires a coordinated effort among personnel from TIRKS, NSDB, OPS/INE, and the Circuit Provisioning Center (CPC). Problems with this interface can be captured in the audit log and resent to OPS/INE via the VJDBAUT format. The log can then be printed to a user–specified printer.

*Table Maintenance*

When the OPS/INE interface is being turned up, you will need to populate several new tables, including the ODB LOCATIONS TTS table. The ODB LOCATIONS table defines which locations support the OPS/INE interface. It is important to change the INE SUPPORTED flag on the analogous TIRKS format, CMOPT, from Y (Yes) to N (No) to prevent TIRKS from passing the same data to OPS/INE.

The ODB ROUTEBY TTS table deserves special mention as well. This table currently gives users the ability to route circuit information to multiple OPS/INE machines through the use of aggregate city/state and aggregate state routeby criteria. Most BCCs, however, will encounter only one OPS/INE machine per NSDB machine in the initial turnup of this interface. To save processing time, it is suggested that the NSDB system administrator use either the TIRKS SEC code or the city/state and state routeby criteria in the ODB ROUTEBY TTS table.

## 6.10 WFA/C-NSDB-ITS/ISDN Interface

### 6.10.1 Description

*Purpose*

The purpose of this interface is to provide WFA/C users with access to the Auto–test feature of ITS for ISDN circuits.

*Data Source*

NSDB receives test requests from WFA/C, manages a worklist containing these requests, and sends the ISDN circuit data associated with the test request to ITS. When ITS has completed the testing, the test results are sent to NSDB. These test results are then passed to WFA/C. The test request is removed from the worklist in NSDB when a completion or cancellation message is received from WFA/C.

*TCM Paths*

Three TCM paths are required between NSDB and ITS/ISDN for this interface. The pathname NSDBTOS is used twice – once from NSDB to ITS, and once from ITS to NSDB. An external acknowledgement message from ITS to NSDB uses the pathname ITSTCMAK. The link from NSDB to WFA/C is not a TCM path, so no TCM paths are required between NSDB and WFA/C.

### 6.10.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. The list below displays the required and optional tables and screens necessary to support the WFA/C-NSDB-ITS/ISDN interface. The WFA/C, TCM, and ITS tables are normally set up through a cooperative effort between the NSDB, WFA/C, and ITS system administrators and communications administrators. For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired.

- NSDB tables:

  ODB CO OPTIONS
  ODB DATA ROUTING
  ODB EXCEPTIONS
  ODB EXK TO SW ID
  ODB SEC TO SRC
  ODM TOS TO EXK

- TCM tables and screens (for the NSDB side):

  TCM USER CONTROL

  TCM Network Administration screens, VJMPNET*
  3 required per logical interface to ITS:
  send to ITS
  receive from ITS
  external acknowledgement from ITS

### 6.10.3 Maintaining the Interface

The status of ITS/ISDN trouble reports can be viewed by using the Work Request Details format (VJDMWD) or through the use of the NSDB Database Modifier (VJDIDBM).

* Each ITS system served requires its own set of send, receive, and acknowledgement TCM Network Administration screens.

## 6.11 WFA/DO-NSDB-ITS/ISDN Interface

### 6.11.1 Description

*Purpose*

The purpose of this interface is to provide WFA/DO users with access to the Auto–test feature of ITS for ISDN circuits.

*Data Source*

NSDB receives test requests from WFA/DO, manages a worklist containing these requests, and sends the ISDN circuit data associated with the test request to ITS. When ITS has completed the testing, the test results are sent to NSDB. These test results are then passed to WFA/DO. The test request is removed from the worklist in NSDB when a completion or cancellation message is received from WFA/DO.

*TCM Paths*

Six TCM paths are used for this interface:

- NSDB receives test requests from WFA/DO on the pathname ADDTST.
- Cancels or completions from WFA/DO are received on pathname CANCMP.
- NSDB sends results to WFA/DO on pathname SNDRSL.
- NSDB sends test requests to ITS/ISDN on pathname NSDBTOS (scenario type A).
- NSDB receives test results from ITS/ISDN on pathname NSDBTOS (scenario type Z).
- ITS/ISDN acknowledgements to NSDB are received on pathname ITSTCMAK.

### 6.11.2 Setting Up the Interface

The NSDB System Administrator must create and maintain several tables and screens to support this interface. The list below displays the required and optional tables and screens necessary to support the WFA/DO-NSDB-ITS/ISDN interface. For the administrative procedures necessary to set up the tables, refer to the sections that describe the table or screen desired.

- NSDB tables:

  ODB CO OPTIONS
  ODB DATA ROUTING
  ODB EXCEPTIONS
  ODB EXK TO SW ID
  ODB SEC TO SRC
  ODM TOS TO EXK

- TCM tables and screens (for the NSDB side):

  TCM USER CONTROL

  TCM Network Administration screens, VJMPNET*
  Minimum total of 6 required per logical interface to ITS:
  send to ITS (1)
  receive from ITS (1)
  receive acknowledgement from ITS (1)
  PLUS a set of the following for each external system (e.g., WFA/DO):
  receive from external system (2)
  send to external system (1)

### 6.11.3 Maintaining the Interface

The status of ITS/ISDN trouble reports can be viewed by using the Work Request Details format (VJDMWD) or through the use of the NSDB Database Modifier (VJDIDBM).

* Each ITS system served requires its own set of send, receive, and acknowledgement TCM Network Administration screens.

# The NSDB Tables
# CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# 7. The NSDB TTS Tables

There are 45 NSDB and subsystem tables that the System Administrator uses to select and map the information necessary for communication with the supported systems. These tables provide flexibility and allow for growth in the dynamic relationship between NSDB and the Provisioning and Operations systems.

For a complete list of the required tables for a given interface, see the "Quick Reference Guide to the Tables for NSDB Interfaces" in chapter 6.

The NSDB tables are listed below:

A/R AFD (refer to section 9.8 "Archive and Retrieval Capabilities for NSDB")
A/R RFD (refer to section 9.8 "Archive and Retrieval Capabilities for NSDB")
DYNAMIC ALLOCATE
MAIL EXITS
MAIL GROUPS
ODB AUDIT
ODB BANKCODES
ODB CO OPTIONS
ODB CTVAL
ODB DATA ROUTING
ODB DATAPORT
ODB DEBUG GLOBAL
ODB DEBUG TRANS
ODB DEFAULT TAGS
ODB EXCEPTIONS
ODB EXEC PARM
ODB FAC CKT DB
ODB HECIG TO EQP
ODB IMAGE
ODB LOCATIONS
ODB MCO OPTIONS
ODB MON SWITCH
ODB OFFICE DATA
ODB ROUTEBY
ODB SCVAL
ODB SDT DEFAULTS
ODB SEC TO SRC
ODB SERVICE CODE
ODB USOC OPTIONS
ODB VALID FORMAT
ODB WCLLI – WCTR

ODM EXK TO SW ID
ODM TOS TO EXK
OQS DB2 TABLES
OQS USERIDS
S1 EXITS
SOAC TO SOPID
TCM SCHEDULE
TCM USER CONTROL
TIME ZONE OPTION
TTS SECURITY
TTS SUBSYSTEMS
TTS VALID XREF
ZONE NAMES
ZONE OFFSETS

## 7.1 A Guide to NSDB Table Keys and Table Record Keys

The following tables list the NSDB TTS tables and the associated Table Keys and Table Record Keys.

**Table 7–1.** A Guide to NSDB Table Keys and Table Record Keys (1 of 3)

| TABLE NAME | TABLE KEY | TABLE RECORD KEY |
|---|---|---|
| DYNAMIC ALLOCATE | none | none |
| MAIL EXITS | none | none |
| MAIL GROUPS | none | none |
| ODB AUDIT | none | none |
| ODB BANKCODES | INE | none |
| ODB CO OPTIONS | none | none |
| ODB CTVAL | ALLMCO<br>MCO Codes | none |
| ODB DATA ROUTING | INE<br>NMAFAC<br>NMASWI<br>NMAISDN<br>(INQWRK)<br>ITS–ISDN SEC Code<br>(ADDTST) WFA Center<br>WFA | none<br>State or City & State CLLI<br>State or City & State CLLI<br>State or City & State CLLI<br>ISDN<br><br><br>6–digit Exchange Key<br><br>SNDMTH, SNDRSL, SNDSWP, UNSSO |
| ODB DATAPORT | INE | none |
| ODB DEBUG GLOBAL | none | DEFAULT or IMS user ID |
| ODB DEBUG TRANS | none | Transaction name (e.g., VJDBL4T) |
| ODB DEFAULT TAGS | P (Packet) or<br>V (Voice) | Exchange Key or<br>Switch Type |
| ODB EXCEPTIONS | Admin Area<br>ALL<br>MCO | From MSG Number |
| ODB EXEC PARM | none | none |

**Table 7-1.** A Guide to NSDB Table Keys and Table Record Keys (2 of 3)

| TABLE NAME | TABLE KEY | TABLE RECORD KEY |
|---|---|---|
| ODB FAC CKT DB | none | none |
| ODB HECIG TO EQP | INE for OPS/INE, but none for other interfaces | HECIG Code Values |
| ODB IMAGE | none | none |
| ODB LOCATIONS | INE | none |
| ODB MCO OPTIONS | none | MCO Code Values<br>Admin Area Values |
| ODB MON SWITCH | none | GTAS Universe Codes for Switch Types |
| ODB OFFICE DATA | none | Central Office CLLI Code Values |
| ODB ROUTEBY | none | INE<br>NMAFAC<br>NMAISDN<br>NMASWI |
| ODB SCVAL | MCO<br>ALLMCO | MCO CLLI Values |
| ODB SDT DEFAULTS | Exchange Key or Switch Type | Tag name from ODB DEFAULT TAGS table |
| ODB SEC TO SRC | none | System Entity Codes (SEC Values) |
| ODB SERVICE CODE | none | DDS Service Code Values<br>ISDN Service Code Values |
| ODB USOC OPTIONS | none | Class of Service USOCs |
| ODB VALID FORMAT | none | Valid Format Names |
| ODB WCLLI – WCTR | none | Central Office CLLI Code Values |
| ODM EXK TO SW ID | none | Exchange Key Values |
| ODM TOS TO EXK | none | ISDN |
| OQS DB2 TABLES | none | OQS database name that maps to a DB2 table |
| OQS USERIDS | Refer to 190-534-003 | Refer to 190-534-003 |

**Table 7-1.** A Guide to NSDB Table Keys and Table Record Keys
(3 of 3)

| TABLE NAME | TABLE KEY | TABLE RECORD KEY |
|---|---|---|
| S1 EXITS | not used | not used |
| SOAC TO SOPID | none | SOAC SEC from ROUTECTL TCM message |
| TCM SCHEDULE | none | none |
| TCM USER CONTROL | none | none |
| TIME ZONE OPTION | none | none |
| TTS SECURITY | Refer to 190-534-003 | Refer to 190-534-003 |
| TTS SUBSYSTEMS | Refer to 190-534-003 | Refer to 190-534-003 |
| TTS VALID XREF | none | none |
| ZONE NAMES | none | Zone Names |
| ZONE OFFSETS | none | Time Zone IDs |

## 7.2 DYNAMIC ALLOCATE

### 7.2.1 Table Description

The DYNAMIC ALLOCATE TTS table contains the parameters to create a debug dataset.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  DYNAMIC ALLOCATE  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: PARMS FOR DEBUG/REPORT DATASETS            REL/LEV:          MOD:
   FIELD     FIELD
   NAME      VALUE
   DISKUNIT          *
   VIO_UNIT          *
   LRECL        *
   BLKSIZE         *
```

**Figure 7–1.** DYNAMIC ALLOCATE Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | Give the actual member name assigned to the tabbed dataset. |
| **Associated Subsystem:** | SSC |
| **Related Table(s):** | None |
| **Format(s):** | None |
| **Module(s):** | None |

### 7.2.2 Field Descriptions

DISKUNIT

| | |
|---|---|
| Full Name: | Disk (DASD) Unit Name |
| Definition: | The unit type or parameter for the debug dataset when debug writes to a disk. |
| Data Content: | up to 8 A/N |
| Valid Entries: | A valid DASD device type or group name (contact ISO). "DISK" is the default. |

VIO_UNIT

| | |
|---|---|
| Full Name: | Esoteric Alias Virtual Unit Name |
| Definition: | The unit type for the debug dataset when debug writes to a virtual dataset. This field is used when the Esoteric Alias Virtual Unit name is not the literal "VIO". |
| Data Content: | up to 8 A/N |
| Valid Entries: | "VIO" is the default. This field is blank if the Esoteric Alias Virtual Unit name equals "VIO". Otherwise, it is populated with the name defined in the MVS GEN (contact the ISO organization for the alias name). |
| Examples: | SYSDA, DASD, 3390, or 3380 |

LRECL

Full Name: Logical Record Length

Definition: The logical record length desired for the debug dataset.

Data Content: 3 N

Valid Entries: Any numeric combination.
"81" is the default.

BLKSIZE

Full Name: Block Size

Definition: Defines the block size for the debug dataset.

Data Content: 5 N

Valid Entries: Any numeric combination.
"00000" is the default.

**NOTE** – If SMS is not being used, the block size must equal or exceed the LRECL.

Examples: 00000 (if DASD is managed by SMS, an IBM product)
09600
27998

### 7.2.3 Creating the Table

**NOTE** – Consult your ISO organization for help in building this table.

1. Display the TTS table access screen by typing /FOR VJTDSO and pressing the ENTER key.
2. On the returned screen, type "DYNAMIC ALLOCATE" in the TABLE NAME field.
3. Leave the TABLE KEY and TABLE RECORD KEY fields blank.
4. Perform a FIND by pressing the PF1 function key.
5. Enter the appropriate field values.
6. Press the PF4 function key to add the record.

### 7.2.4 Maintaining the Table

1. Display the TTS table access screen by typing /FOR VJTDSO and pressing the ENTER key.
2. On the returned screen, type "DYNAMIC ALLOCATE" in the TABLE NAME field.
3. Leave the TABLE KEY and TABLE RECORD KEY fields blank.
4. Perform a FIND by pressing the PF1 function key.

- To update an entry, enter a C to the left of the field being modified, make the appropriate modifications, and press the PF5 function key.
- To delete the record, press the PF10 function key.

## 7.3 MAIL EXITS

### 7.3.1 Table Description

This table contains core software that may be used in the future.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  MAIL EXITS        TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0000
NOTE: INVOKE MESSAGE SWITCH TO THIS TRANSACTIO REL/LEV:            MOD: Y
   FIELD     FIELD
   NAME      VALUE
   PRODLINE    *
   PGMNAME           *
   LANGUAGE     *
```

**Figure 7–2.** MAIL EXITS Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore (locked table) |
| **Updating Cautions:** | None |
| **Associated Subsystem:** | SSC |
| **Format(s):** | VJMAIL |
| **Module(s):** | VJSSMWC, VJSSMRC |

### 7.3.2 Field Descriptions

This table is not currently used.

### 7.3.3 Creating the Table

This table is not currently used.

### 7.3.4 Maintaining the Table

This table is not currently used.

## 7.4 MAIL GROUPS

### 7.4.1 Table Description

This table contains a list of User IDs, Logical Terminals (LTERMS), or groups that are to receive broadcast or bulk electronic mail.

The format used to send and receive mail is VJMAIL. For more information on electronic mail, see Bellcore BR 190-534-311.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  MAIL GROUPS       TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS:
NOTE: TBL OF MAIL GROUPS FOR BROADCAST FUNC    REL/LEV:           MOD: Y
   FIELD    FIELD
   NAME     VALUE
   MAIL_ID          *
   ID_TYPE   *
   MAILNM01         *
   MAILNM02         *
   MAILNM03         *
   MAILNM04         *
   MAILNM05         *
   MAILNM06         *
   MAILNM07         *
   MAILNM08         *
   MAILNM09         *
   MAILNM10         *
   MAILNM11         *
   MAILNM12         *
   MAILNM13         *
```

**Figure 7–3.** MAIL GROUPS Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  MAIL GROUPS       TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: TBL OF MAIL GROUPS FOR BROADCAST FUNC     REL/LEV:           MOD:
   FIELD    FIELD
   NAME     VALUE
   MAILNM14          *
   MAILNM15          *
   MAILNM16          *
   MAILNM17          *
   MAILNM18          *
   MAILNM19          *
   MAILNM20          *
   MAILNM21          *
   MAILNM22          *
   MAILNM23          *
   MAILNM24          *
   MAILNM25          *
```

**Figure 7–4.** MAIL GROUPS Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Associated Subsystem:** | SSC |
| **Format(s):** | VJMAIL |
| **Module(s):** | VJSSMRP |

### 7.4.2 Field Descriptions

MAIL_ID

Full Name: Mail Group Identification

Definition: Identifies the name of a group, or groups, that will receive messages by electronic mail.

Data Content: 8 A/N

Valid Entries: Any valid user ID or LTERM ID of 8 characters or less.

ID_TYPE

Full Name: Identification Type

Definition: Identifies the type of group that will receive messages by electronic mail.

Data Content: 1 A

Valid Entries: U = Group of User IDs
L = Group of LTERMS
G = Group of Groups

MAILNM01

Full Name: Mail Name 01 through Mail Name 25

Definition: Identifies the names of group members, LTERMS, or groups that will receive messages by electronic mail.

Data Content: 25 fields, maximum of 8 A/N characters each.
KLSSCDJW (e.g., of user ID)
PY452006 (e.g., of LTERM)

### 7.4.3 Creating the Table

See BR 190-534-311 for details.

### 7.4.4 Maintaining the Table

See BR 190-534-311 for details.

## 7.5 ODB AUDIT

### 7.5.1 Table Description

The ODB AUDIT TTS table is used to control logging and deletion processing for the NSDB audit process. This table controls the data associated with the VJDBAUT screen.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB AUDIT          TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: ODB AUDIT                                  REL/LEV:            MOD: Y
   FIELD    FIELD
   NAME     VALUE
   DLT_CMPL N*
   AUDIT_ON Y*
   BLD_ARC  Y*
   ERRORS   Y*
   NO_ERR#   769                                                        *
   NO_ERR#2                                                             *
   NO_SNAP#                                                             *
   NO_ETBL#                                                             *
   NO_CTBL#                                                             *
   SUCCESS  Y*
   SCS_CCAC Y*
   SCS_MCAC Y*
   SCS_NCAC Y*
   SCS_SCAC Y*
   BLKPROC# 000000150*
TTS206I FIND COMPLETED  CONTINUED ON NEXT PAGE
```

**Figure 7–5.** ODB AUDIT Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB AUDIT          TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0001
NOTE: ODB AUDIT                                   REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   BLKDELAY 000000000*




TTS210I FIND COMPLETED
```

**Figure 7–6.** ODB AUDIT Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.1 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Format(s):** | VJDBAUT |
| **Module(s):** | VJDBA1P, VJDBA2P, VJDBALM, VJDBAUF, VJDBAUM, VJDBBAT, VJDBL1P, VJDBL4P, VJDBL5P, VJDBL6P, VJDBT2P |

### 7.5.2 Field Descriptions

DLT_CMPL

Full Name: Delete Completions

Definition: Indicates whether or not messages related to a circuit activity will be deleted when a circuit activity is completed. If the DLT_CMPL flag is set to Y, the BLD_ARC flag must be set to N.

**NOTE** – This is an online DELETE process.

Data Content: 1 A

Valid Entries: Y = Yes, delete circuit activity information at completion. Any updates to an in-effect circuit after completion WILL NOT be deleted. Any error messages generated without a circuit ID and order number WILL NOT be deleted.

N = No, do not delete circuit activity information at completion. N is the default value.

AUDIT_ON

Full Name: Audit On

Definition: Indicates whether or not the audit feature is turned on; determines whether the other fields on the ODB AUDIT table are used.

Data Content: 1 A

Valid Entries: Y = Other fields on the ODB AUDIT table will be used to determine what to log and/or delete. Y is the default value.

N = No, logging and deletions at completion will not be performed. A value of N causes all other fields on the ODB AUDIT table to be ignored.

BLD_ARC

Full Name: Build Archive Audit Record

Definition: This flag indicates whether an archive audit record should be built at the completion of a circuit or when an in–effect circuit is updated. If the generic batch archive process will be used to archive the Audit tables, then the BLD_ARC field must be set to "Y". This will cause an audit archive record to be added to the Alias database. If the generic batch archive process will *not* be used, then the BLD_ARC field should be set to "N". This will prevent the audit archive record from being added to the Alias database. If the DLT_CMPL flag is set to Y, the BLD_ARC flag must be set to N. The default is "Y".

Data Content: 1 A

Valid Entries: Y or blank = Yes, the generic batch archive process will be used to archive or remove the contents of the Audit tables.

N = No, the generic batch archive process will *not* be used to archive or remove the contents of the Audit tables.

ERRORS

Full Name: Error and Exception messages

Definition: This field controls the logging of error and exception messages.

Data Content: 1 A

Valid Entries: Y = Yes, errors and exceptions will be logged to the Audit database. Y is the default value.

N = No, errors and exceptions will not be logged to the Audit database.

NO_ERR#

Full Name: No Error Number

Definition: Indicates the error numbers that will *not* be logged even if the ERRORS field is set to Yes. To suppress error and exception numbers for a specific contract, concatenate the number and the contract. For example, to suppress 301 numbers for the Update contract, you could enter "301UPDCKT" or "UPDCKT301".

Data Content: 59 A/N (Error codes must be separated with at least one blank.)

NO_ERR#2

Full Name: No Error Number #2

Definition: Indicates the error numbers that will *not* be logged even if the ERRORS field is set to Yes. This is a continuation of the NO_ERR# field for the suppression of specified error numbers. Both fields are checked, so error and exception numbers can appear in either field.

Data Content: 59 A/N (Error codes must be separated with at least one blank.)

NO_SNAP#

Full Name: No Snap

Definition: Restricts the logging of a snap message, a quick program trace, for the designated error number(s). Snap messages are usually generated for severe errors, and will be used mainly by Bellcore NSDB developers to solve these errors in a short time frame. If the error number is suppressed in the NO_ERR# field or the NO_ERR#2 field, a snap cannot be obtained under *any* circumstances.

Data Content: 59 A/N (Error codes must be separated with at least one blank.)

NO_ETBL#

Full Name: No Error Table

Definition: Specifies the error numbers which will *not* be logged to the Error/Exception table. This field does not prevent logging to the Circuit tables.

Data Content: 59 A/N (Error codes must be separated with at least one blank.)

NO_CTBL#

Full Name: No Circuit Table

Definition: Specifies the error numbers which will *not* be logged to the Circuit tables.This field does not prevent logging to the Error/Exception tables.

Data Content: 59 A/N (Error codes must be separated with at least one blank.)

SUCCESS

Full Name: Successful transactions

Definition: Indicates whether or not successful transactions should be logged.

Data Content: 1 A

Valid Entries: Y = Yes, log successful transactions to the Audit database. A "Y" indicates that the next four fields (SCS_CCAC, SCS_MCAC, SCS_NCAC, and SCS_SCAC) will be checked to see which circuit types are designated for inclusion. Y is the default.

N = No, do *not* log successful transactions to the Audit database. Do *not* check the next four fields to determine which successful transactions should be logged.

SCS_CCAC

Full Name: Log Successes for Carrier Circuits

Definition: Indicates whether successful transactions should be logged for carrier circuits, based on the first byte of the CAC.

Data Content: 1 A

Valid Entries: Y = Yes, carrier circuits should be logged. Y is the default.

N = No, carrier circuits should *not* be logged.

SCS_MCAC

Full Name: Log Successes for Message Circuits

Definition: Indicates whether successful transactions should be logged for message circuits, based on the first byte of the CAC.

Data Content: 1 A

Valid Entries: Y = Yes, message circuits should be logged. Y is the default.

N = No, message circuits should *not* be logged.

SCS_NCAC

Full Name: Log Successes for Nondesigned Circuits

Definition: Indicates whether successful transactions should be logged for nondesigned circuits, based on the first byte of the CAC.

Data Content: 1 A

Valid Entries: Y = Yes, nondesigned circuits should be logged. Y is the default.

N = No, nondesigned circuits should *not* be logged.

SCS_SCAC

Full Name: Log Successes for Special Circuits

Definition: Indicates whether successful transactions should be logged for special circuits, based on the first byte of the CAC.

Data Content: 1 A

Valid Entries: Y = Yes, special circuits should be logged. Y is the default.

N = No, special circuits should *not* be logged.

BLKPROC#

Full Name: Bulk Processing Number

Definition: Defines the number of errors to bulk process in a single bulk recovery command before stopping. (The default is 100.)

Data Content: 9 N

BLKDELAY

Full Name: Bulk Delay

Definition: BLKDELAY is only effective for the SNDINE, SNDNMA, and SNDWFA Bulk commands. It defines the time interval to delay before processing the next error handled by the bulk process. This delay does not require CPU or force U240 abends. BLKDELAY is useful when sending data to other systems, such as the NMA system. This option prevents the downstream system from being flooded by messages. (The default is *no* delay.)

Data Content: 9 N (expressed in milliseconds; i.e.,1000 = 1 second = 1 CPU second)

Example: 2000 (2 CPU seconds)

### 7.5.3 Creating the Table

The NSDB System Administrator should create this table with an audit plan in mind. The ODB Audit table can accomodate audit logging of all or selected messages, and can delete messages upon completion of an order.

Follow the steps below to create the ODB AUDIT table:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB AUDIT in the TABLE NAME field, and press PF1 to FIND it.
3. On the returned screen, type the values that fit your audit needs into the appropriate fields.
4. When these values are filled in, ADD the information to ODB AUDIT by pressing the PF4 function key.

### 7.5.4 Maintaining the Table

This table can be modified as needed to turn on/off the audit feature, delete audits upon completion of an order, selectively exclude error and exception messages from audit logging, and exclude/include success messages from the audit feature.

> **NOTE –** Updates to the ODB AUDIT table will take effect immediately, except for messages already stacked up for the VJDBBAT transaction.

## 7.6 ODB BANKCODES

### 7.6.1 Table Description

The ODB BANKCODES table is used to determine which bank type codes are supported by OPS/INE.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB BANKCODES     TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: NSDB BANKCODE VALIDATION TABLE             REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   BANKCODE        *
```

**Figure 7–7.** ODB BANKCODES Table

| | |
|---|---|
| **Table Key(s):** | INE |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | Bellcore |
| **Updating Cautions:** | All bank codes on this table should be cross–checked with the ODB DATAPORT table when adding or removing bank codes. |
| **Related Table(s):** | ODB DATAPORT |
| **Format(s):** | None |
| **Module(s):** | VJDBICK |

### 7.6.2 Field Descriptions

BANKCODE

| | |
|---|---|
| Full Name: | Bank Type Code |
| Definition: | The code that identifies the network elements that can be provisioned by the OPS/INE system for a TIRKS–generated circuit. |
| Data Content: | 6 A/N |
| Valid Entries: | Any valid bank type code |

### 7.6.3 Creating the Table

This table is prepopulated at Bellcore with the bank type codes supported by OPS/INE. To view data on this table, follow these steps:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB BANKCODES" in the TABLE NAME field.
3. Type "INE" in the TABLE KEY field.
4. Perform a FIND by pressing the PF1 function key.

**NOTE –** The bank type codes in this table are equivalent to values in the CIMAP DACSBANKS and INE CHNBANKS TTS tables presently stored in the TIRKS system, and may be copied over at the initial turnup of the OPS/INE interface.

### 7.6.4 Maintaining the Table

The ODB BANKCODES table is prepopulated with data, and is maintained by Bellcore.

## 7.7 ODB CO OPTIONS

### 7.7.1 Table Description

The Operations Data Base Company Options (ODB CO OPTIONS) table is used to support each of the interfaces shown above. This table performs several functions. These functions include:

- Identification of the Operations systems that NSDB communicates with.
- Selection of a default MCO.
- Identification of the error message Logical Terminal (LTERM).
- Selection of various processing options.
- Identification of bridge numbering conventions for the TIRKS-NSDB-ITS/Digital interface.
- An option to turn the NSDB–OPS/INE link on or off for issues and reissues of the WORD document from the TIRKS system, and an option to turn the NSDB–OPS/INE link on or off for disconnect orders at the time of logging in GOC.
- Determination of whether or not to call the debugging modules for transactions that do not support a screen use.
- Options to indicate whether or not the Facility–to–Circuit databases are installed.
- An option to turn on default translation generation for ISDN services.
- Options to stop the Merge Line Record (MLR) process which rebuilds database partitions or to stop bulk error processing.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB CO OPTIONS    TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0001
NOTE: ODB COMPANY OPTIONS                          REL/LEV:           MOD: Y
   FIELD  :  FIELD
   NAME      VALUE
   DFLT_MCO DEFAULT    *
   MSGLTERM PY411BJ1*
   FLTRLOAD N*
   PORTNAM1 PT##           *
   PORTNAM2 PRT##          *
   PORTNAM3 BR##           *
   ITS_COMP  *
   VALIDATE Y*
   LOAD_IP  Y*
   BLD_TE   Y*
   CHK_SVC  Y*
   CHK_RP   Y*
   SSC_ON   Y*
   SSC_LINK         *
   ITS_ON   Y*
TTS206I FIND COMPLETED  CONTINUED ON NEXT PAGE
```

**Figure 7–8.** ODB CO OPTIONS Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB CO OPTIONS    TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0001
NOTE: ODB COMPANY OPTIONS                         REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   NMA_ON    Y*
   NMAI_ON   Y*
   INE_ON    Y*
   WFA_ON    Y*
   DISC_OL   Y*
   NSDB_SEC  OPDB1SEC*
   DEBUG_ON  Y*
   COS_PKT   N*
   MANUALWO  Y*
   USOC_OPT  Y*
   C_NO_S&E  Y*
   FACCKTDB  Y*
   ISDN_DEF  B*
   MISCDATA  BAT=NONE                                                  *

TTS210I FIND COMPLETED
```

**Figure 7–9.** ODB CO OPTIONS Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBIQF, VJDBL1P, VJDBL2P, VJDBL3P, VJDBL4P, VJDBL5P, VJDBL6P, VJDBMLR, VJDBNO1, VJDBR1P, VJDBUPF |

### 7.7.2 Field Descriptions

DFLT_MCO

Full Name: Default Maintenance Control Office

Definition: The CLLI code for the Default MCO. This field is used when the CKT MCO on the trouble report is blank.

Data Content: 11 A/N

Valid Entries: A valid MCO. Must be an MCO that is contained in the Partition Matrix NSDB database.

MSGLTERM

Full Name: Message Logical Terminal

Definition: Identifies the Logical Terminal (LTERM) at a MCO that is the destination of error messages. This value is assigned by the BCC.

Data Content: 8 A/N

Example: PY511CG2

FLTRLOAD

Full Name: Filter Load

Definition: A flag indicating whether to filter out the circuit details that are not in the Partition Matrix database records.

Data Content: 1 A/N

Valid Entries: Y (Yes, filter out the circuit details that contain MCOs.)

N (No, load the circuit details into the default partition.)

PORTNAM1

Full Name: Bridge Assignment Parsing Mask No. 1

Definition: This field, along with PORTNAM2 and PORTNAM3, is used to define to NSDB the port naming conventions found on WORD documents processed by NSDB. This field is required because of variations in naming conventions found in the existing and evolving pool of records encountered by NSDB. The entries made here are critical to the success of the multi-point mapping process. See the NOTE below for more information.

Data Content: Up to 15 A/N

Valid Entries: PT ##
PORT ###
PRT ###

**NOTE –** The number of digits in the port number must correspond to the number of digits required for a particular port name (e.g., PT ## must have a 2-digit port number). If the number of "#" signs is less than the number of digits in the port number encountered, then the actual port number will be truncated by the NSDB Data Base Manager and the multi-point mapping will be built incorrectly. The values entered for PORTNAM1, PORTNAM2, and PORTNAM3 should be different and should list the most likely possibilities that NSDB will encounter.

PORTNAM2

Full Name: Bridge Assignment Parsing Mask No. 2

Definition: (See definition under PORTNAM1.)

Data Content: 15 A/N

Valid Entries: PT ##
PORT ###
PRT ###

PORTNAM3

Full Name: Bridge Assignment Parsing Mask No. 3

Definition: (See definition under PORTNAM1.)

Data Content: 15 A/N

Valid Entries: PT ##
PORT ###
PRT ###

ITS_COMP

Full Name: Integrated Test System (ITS) Compression Flag

Definition: A flag indicating whether or not NSDB should compress the data being sent to ITS.

Data Content: 1 A

Valid Entries: Y (Yes) This is the entry used under normal conditions

N (No) This entry is only used for certain debugging conditions

VALIDATE

Full Name: Validate Access Point Data Flag

Definition: Indicates whether or not access point data (from the circuit details section of the WORD document) requires validation. See the NOTE below for possible impacts of this validation.

**NOTE –** If Y is entered, NSDB will NOT store any specials, carrier, or message circuits that have invalid access point data. If N is entered, NSDB WILL store all circuits regardless of the condition of the access point data. *For BCCs that use NSDB only as an interface to ITS, this entry should be Y, since ITS requires a test point.* If the BCC has an interface with the NMA system, then the entry should be N to ensure that all appropriate circuits are stored by NSDB.

Data Content: 1 A

Valid Entries: Y (Yes, NSDB will validate access point data.)

N (No, NSDB will not validate access point data.)

LOAD_IP

Full Name: Load Index Page Flag

Definition: Indicates whether or not a BCC is currently using NSDB to receive and store the index page from the TIRKS system. The option Y permits TIRKS–to–NSDB multipoint index page processing for systems such as OPS/INE..

Data Content: 1 A

Valid Entries: Y (Yes)

N (No)

BLD_TE

Full Name: Build Test Point Equipment Flag

Definition: Indicates whether or not NSDB will perform special processing when test point equipment is encountered on the circuit details section of the WORD document.

Data Content: 1 A

Valid Entries: Y (Yes, special processing will take place. This includes a check of the ODB HECIG TO EQP table for a test equipment HECIG. Use Y if NSDB supports ITS and the NMA system is not deployed.)

N (No, special processing should not take place. Use N if the NMA system has an interface with NSDB.)

CHK_SVC

Full Name: Check Service Code Flag

Definition: A flag indicating whether or not NSDB should check the ODB SERVICE CODE table each time a special services circuit is issued via the WORD document.

Data Content: 1 A

Valid Entries: Y (Yes, NSDB will check the service code encountered on the WORD document and compare it to the service codes in the ODB SERVICE CODE table to determine if the circuit is a DDS circuit. Use Y if NSDB has an interface with ITS.)

N (No, the service code will not be checked. Use N if NSDB has an interface with the NMA system *only*.)

CHK_RP

Full Name: Check Repeater Plug-in Flag

Definition: A flag indicating whether or not NSDB will check the ODB HECIG TO EQP table for verification that an encountered plug-in is of the type used for 56Kb/sec DDS circuits.

Data Content: 1 A

Valid Entries: Y (Yes, NSDB should check for repeater plug-ins. Use Y if NSDB has an interface to ITS.)

N (No, NSDB will not check for repeater plug-ins. Use N if NSDB has an interface to the NMA system *only*.)

SSC_ON

Full Name: NSDB to CIMAP/SSC Interface Flag

Definition: A flag indicating whether or not the NSDB to CIMAP/SSC interface exists.

Data Content: 1 A

Valid Entries: Y (Yes, the interface exists.)

N (No, the interface does not exist.)

SSC_LINK

Full Name: CIMAP/SSC to NSDB Multiple System Coupling (MSC) Link Name

Definition: Identifies the MSC link name used for the interface between CIMAP/SSC and NSDB. This field is used only if there is an interface to CIMAP/SSC. If NSDB is deployed with WFA/C in the same IMS control region, then this field is not used.

Data Content: 8 A/N

Valid Entries: Blank or a BCC–provided valid MSC link name.

ITS_ON

Full Name: NSDB to ITS Analog Interface Flag

Definition: A flag indicating whether or not the ITS Analog interface to NSDB exists. *This field is not currently used.*

Data Content: 1 A

Valid Entries: Y (Yes, the interface exists.)

N (No, the interface does not exist.)

NMA_ON

Full Name: NSDB to NMA Interface Flag

Definition: A flag indicating whether or not the NMA interface to NSDB exists.

Data Content: 1 A

Valid Entries: Y (Yes, the interface exists.)

N (No, the interface does not exist.)

NMAI_ON

Full Name: NMA/ISDN Flag

Definition: A flag indicating whether or not NMA/ISDN is in use.

Data Content: 1 A

Valid Entries: Y (Yes, NMA/ISDN is in use.)

N (No, NMA/ISDN is not in use.)

INE_ON

Full Name: OPS/INE Flag

Definition: A flag indicating whether or not the NSDB–OPS/INE link for issues and reissues of the WORD document is in use.

Data Content: 1 A

Valid Entries: Y (Yes, the NSDB–OPS/INE link for the WORD is on)

N (No, the NSDB–OPS/INE link for the WORD is not on)

WFA_ON

Full Name: NSDB to WFA Interface Flag

Definition: A flag indicating whether or not the WFA interface to NSDB exists.

Data Content: 1 A

Valid Entries: Y (Yes, the interface exists.)

N (No, the interface does not exist.)

DISC_OL

Full Name: Disconnects on Logging

Definition: A flag indicating whether OPS/INE should receive disconnects at the time of logging in GOC.

Data Content: 1 A

Valid Entries: Y (Yes, the NSDB–OPS/INE link for disconnects on logging is on)

N (No, the NSDB–OPS/INE link for disconnects on logging is not on)

NSDB_SEC

Full Name: Network and Services Data Base System Entity Code

Definition: The NSDB SEC code is used to identify NSDB to other products and is also used in TCM administration.

Data Content: 8 A/N

Valid Entries: SEC value for NSDB

DEBUG_ON

Full Name: Debugging Switch

Definition: This flag determines whether or not to call the debugging modules for transactions that do not support a screen use. If the value "N" is typed in, or if this field is left blank, system performance will remain comparable to the performance of prior (pre–3.2) NSDB releases.

Data Content: 1 A

Valie Entries: Y (Yes, access the debugging modules and TTS debugging tables to decide if debugging is necessary.)

N or blank (No, do *not* access the debugging modules and TTS debugging tables to decide if debugging is necessary; N is the default)

COS_PKT

Full Name: COSMOS Packet

Definition: This field indicates whether or not COSMOS packet assignments are to be converted to packet services for ISDN pipes.

Data Content: 1 A

Valid Entries: Y (Yes, convert the COSMOS packet assignment to packet services for ISDN pipes.)

N (No, do not convert the COSMOS packet assignment to packet services for ISDN pipes.)

Default value is Y. Only N will cause no conversion.

MANUALWO

Full Name: Manual Work Order Indicator

Definition: Indicates whether or not the Match Failure database in NSDB is to be turned on. This database stores unmatched SOAC Work Order messages along with their candidate circuit IDs that were found in the NSDB Line Record database.

Data Content: 1 A

Valid Entries: Y (Yes, turn on the Match Failure database for use with manual work order processing.)

N (No, do not turn on the Match Failure database.)

USOC_OPT

Full Name: USOC Options

Definition: Indicates whether or not the ODB USOC OPTIONS TTS table should be called before processing service orders from the SOAC system.

Data Content: 1 A

Valid Entries: Y (Yes, NSDB should access the ODB USOC OPTIONS TTS table.)

N (No, NSDB should not access the ODB USOC OPTIONS TTS table.)

C_NO_S&E

Full Name: Change Order with No Service & Equipment section

Definition: Indicates the processing method to be used when NSDB encounters a Change order from SOAC that does *not* contain an S&E section.

Data Content: 1 A

Valid Entries: Y (Yes, NSDB should process the order using the Main/Miscellaneous Account Number as sent by SOAC in the CTID field. WFA/C will also log the order.)

N (No, NSDB should not process a Change order without an S&E section. This selection will result in an application error back to TCM.)

FACCKTDB

Full Name: Facility–to–Circuit Database Flag

Definition: Indicates whether or not the Facility–to–Circuit databases are installed.

Data Content: 1 A

Valid Entries: Y (the Facility–to–Circuit databases are installed)

N (the Facility–to–Circuit databases are *not* installed; 'N' is the default)

ISDN_DEF

Full Name: ISDN Defaults

Definition: An option to turn on default translation generation for ISDN services.

Data Content: 1 A

Valid Entries: P (set defaults for Packet)

V (set defaults for Voice)

B (set defaults for both Packet and Voice)

N or blank (No, don't set defaults)

MISCDATA

Full Name: Miscellaneous Data

Definition: This field can be used for three purposes:

- to stop the Merge Line Record (MLR) process which rebuilds database partitions
- to stop bulk error processing
- for future use in conjunction with emergency software releases

Data Content: up to 59 A/N

Valid Entries: STOPMLR (Stop the Merge Line Record (MLR) process)

STOPA2P (Stop bulk error processing)

### 7.7.3 Creating the Table

The entries made during the creation of the ODB CO OPTIONS table must accurately reflect the environment that NSDB will encounter upon deployment. The NSDB System Administrator must identify several characteristics about the systems that NSDB will be communicating with, as well as the appropriate options to select on a company level.

Follow the steps below to set up this table:

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB CO OPTIONS in the TABLE NAME field. Press the PF1 function key.
3. On the returned screen, type the appropriate values for each displayed field on the table. (This table has two pages. You must enter the field values on page one before proceeding to page two. Use the PF4 function key to ADD the data on page one. Next, press PF2 to advance to page two. Type in the values on page two and again press the PF4 function key.)

### 7.7.4 Maintaining the Table

The ODB CO OPTIONS table controls numerous on/off options associated with the interfaces that NSDB supports. This table also provides for some features within the interfaces.

The NSDB System Administrator should modify this table as required to support interface turn-ups and options within certain interfaces, and to perform debugging tasks.

Changes and modifications to the ODB CO OPTIONS table can have company-wide impact. Care should be taken to avoid unwanted results when making changes to this table.

## 7.8 ODB CTVAL

### 7.8.1 Table Description

This table is used by NSDB to select the circuit type values that are used in the online validation process. The online validation process is a check of the TIRKS system WORD document for usability by ITS for testing. If the criteria for usability is not met, a message can be immediately sent to a designated LTERM so that the WORD document can be reissued after corrections are made.

The ODB CTVAL table provides an on/off capability for selected circuit types. The System Administrator indicates that online validation should occur by entering Y in the fields associated with special services circuits (SPEC_SVC), and carrier circuits (CARRIER). The fields shown for message circuits (MESSAGE), nondesigned circuits (NON_DSGN), and ISDN circuits (ISDN), are reserved for future use.

This table can be used in conjunction with the ODB SCVAL table to further refine the circuits within a type that are subject to online validation edits.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB CTVAL          TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0001
NOTE: ODB CKT VAL. PERMISSIONS                    REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   MSGLTERM  PY502B00*
   SPEC_SVC  Y*
   MESSAGE   N*
   CARRIER   Y*
   NON_DSGN  N*
   ISDN      N*
   SPARE1     *
   SPARE2     *
   SPARE3     *
   SPARE4     *
   SPARE5     *
   SPARE6     *
   SPARE7     *

TTS210I FIND COMPLETED
```

**Figure 7–10.** ODB CTVAL Table

| | |
|---|---|
| **Table Key(s):** | ALLMCO or MCO codes |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBB1P, VJDBLVP |

**Table 7–2.** ODB CTVAL Online Validation Entries

| IF THESE VALUES ARE ENTERED, | | THEN THE RESULT IS |
|---|---|---|
| ODB CTVAL | ODB SCVAL | |
| SPEC_SVC = Y | DDS = BLANK<br>= N<br>= Y<br>DS1 = BLANK<br>= N<br>= Y | ALL special services circuits are submitted for online validation*. |
| SPEC_SVC = N | DDS = Y<br>DS1 = BLANK<br>= NO | Only special services circuits that are DDS will be submitted for online validation. |
| SPEC_SVC = N | DS1 = Y<br>DDS = BLANK<br>= NO | Only special services circuits that are DS1 will be submitted for online validation. |
| CARRIER = Y | CLFI = BLANK<br>= N<br>= Y | ALL carrier circuits are submitted for online validation**. |
| CARRIER = N | CLFI = Y | Only carrier circuits identified by CLFI will be submitted for online validation **. |

* ALL special services in this case means *both* DS1 and DDS; analog special services are not included in the validation process.

** CLFI circuits are internal BCC–owned carrier circuits; customers can purchase channels in them.

### 7.8.2 Field Descriptions

MSGLTERM

Full Name: Message Logical Terminal

Definition: The destination identifier of the terminal device where messages are sent for circuits that fail the online validation process.

Data Content: 8 A/N

Example PY507DF2

SPEC_SVC

Full Name: Special Services

Definition: Identifies special services circuits (DDS or carrier).

Data Content: 1 A

Valid Entries: Y (Indicates ALL special services circuit types are included in online validation processing.)

N (Indicates ALL special services circuit types are not included in online validation processing.)

MESSAGE

Full Name: Message

Definition: This field reserved for future use.

CARRIER

Full Name: Carrier

Definition: Identifies the Carrier circuit type.

Data Content: 1 A

Valid Entries: Y (Indicates ALL carrier circuit types are included in online validation processing.)

N (Indicates ALL carrier circuit types are not included in online validation processing.)

NON_DSGN

Full Name: Non-designed

Definition: Reserved for future use.

ISDN

Full Name: Integrated Services Digital Network

Definition: Reserved for future use.

SPARE1 through SPARE7

Full Name: Spare

Definition: Reserved for future use

### 7.8.3 Creating the Table

The NSDB System Administrator can use this table and the ODB SCVAL table to select the circuits that will be used in the online validation process. Both of these tables are keyed by MCO. The ODB CTVAL table provides a selection by circuit type. This is a broad category that includes all circuits of the selected type. If further discrimination of the circuits in a circuit type is needed, the System Administrator can use the ODB SCVAL table. The combination of entries made on these two tables can reduce the number of circuits submitted to the online validation process so that only those requiring the process will be considered.

To create the ODB CTVAL table follow the steps below.

1. From a clear screen, type /FOR VJTDSO and hit enter.
2. On the returned screen, type in ODB CTVAL in the TABLE NAME field.
3. Type MCO in the TABLE KEY field.
4. Press PF1 to perform a FIND.
5. On the returned find screen, type in the LTERM information desired in the MSGLTERM field. (This is the destination of the online validation error messages.)
6. Type a "Y" for Yes next to the circuit types that should be submitted for online validation processing.
7. Press PF4 to ADD the values.

### 7.8.4 Maintaining the Table

The NSDB System Administrator should modify this table if the LTERM for error messages is changed. The MSGLTERM field is used to identify the desired LTERM. Changes may also be required when the circuits subject to validation change. See the chart that identifies the entries and their results for the ODB CTVAL and ODB SCVAL tables.

## 7.9 ODB DATA ROUTING

### 7.9.1 Table Description

This table is described for each INTERFACE ID that it supports. This is necessary because the ODB DATA ROUTING table is used differently and is set up separately for each INTERFACE ID. To find the appropriate description, locate the INTERFACE ID heading for the combination of systems that you are interested in. The table description for that INTERFACE ID follows the heading:

**INTERFACE ID = CRS-NSDB-WFA/C**

The ODB DATA ROUTING table is used to route the Telephone Number swap information, initiated by the Centrex Rearrangement System (CRS), to WFA/C.

This table uses a TABLE KEY of WFA, a ROUTEKEY of SNDSWP (meaning 'send swap'), and the NODE fields contain the System Entity Codes for the WFA/C systems.

**INTERFACE ID = SOAC-NSDB-WFA/C**

Two versions of the ODB DATA ROUTING table are used to route SOAC service order data to Special Service Centers that use WFA/C.

The first version uses a TABLE KEY of WFA, a ROUTEKEY of UNSSO (meaning 'unsolicited service order'), and the NODE fields contain the System Entity Codes for the WFA/C systems.

> **NOTE –** To deploy the OPS/INE interface, a node *must* be populated with the WFA/C System Entity Code (SEC). This node needs to be built whether the provisioning source is TIRKS or SOAC in order for OPS/INE to identify which WFA/C system is to receive OPS/INE data.

A second version of this table is also required. The matching process that NSDB uses to associate the TIRKS system data with its corresponding SOAC data results in a message sent to WFA/C. This message indicates whether or not the match was successful. The distribution of this message to the WFA/C center is controlled by the second version of the ODB DATA ROUTING table. This table uses a TABLE KEY of WFA, a ROUTEKEY of SNDMTH (meaning 'send match'), and the NODE fields contain the SEC codes for the WFA/C systems.

INTERFACE ID = TIRKS-NSDB-NMA/Facilities
= TIRKS-NSDB-NMA/Pass Through
= TIRKS-NSDB-NMA/Switch

NSDB uses this table in conjunction with the ODB ROUTEBY table to distribute selected TIRKS data to specific NMA systems. The ODB DATA ROUTING table maps the specific NMA system to the geographical area it serves. The specific NMA system is identified by its System Entity Code (SEC). The geographical area is identified by the city and state, or state common language codes for the A and Z central offices served by the NMA system. The System Administrator can associate multiple NMA systems to a single area (by city and state, or state alone), or a single NMA system can be associated with multiple areas by creating an ODB DATA ROUTING table for each area.

As of NSDB Release 3.0.3, NMA/Switch was provided with support in NSDB to enable the monitoring of ISDN circuits. NSDB support includes the LINK_ON field in the ODB DATA ROUTING table which can be used to distribute data to NMA machines that monitor ISDN circuits. The ODB DATA ROUTING table has a new Table Key of NMAISDN to identify NMA machines that perform ISDN monitoring. The ODB ROUTEBY table has a new Table Record Key of NMAISDN to identify the routing scheme for NMA systems that need ISDN data.

**INTERFACE ID = TIRKS–NSDB & WFA/C–OPS/INE**

NSDB uses this table to identify the OPS/INE systems that should receive circuit data.

> **NOTE –** To deploy the OPS/INE interface, a node *must also* be populated with the WFA/C System Entity Code (SEC) for a table key value of "WFA" and a table record key value of "UNSSO". This node needs to be built whether the provisioning source is TIRKS or SOAC in order for OPS/INE to identify which WFA/C system is to receive OPS/INE data.

INTERFACE ID = WFA/C-NSDB-ITS/ISDN

NSDB uses this table in a different way for ITS/ISDN routing. The ODB DATA ROUTING table is used in conjunction with two other tables in this case. These tables are called ODM TOS TO EXK and ODM EXK TO SW ID. The elements mapped by these tables for the WFA/C-NSDB-ITS/ISDN interface are shown in the figure below.

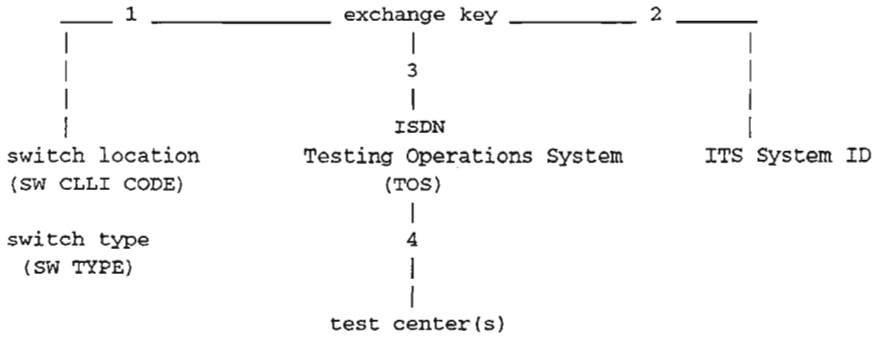



**Figure 7–11.** ODB Data Routing for ITS/ISDN

1. The ODM EXK TO SW ID table associates an exchange key to the switch location and switch type.
2. The ODB DATA ROUTING table associates the same exchange key to the ITS system that tests the ISDN circuits residing within the exchange key.
3. The ODM TOS TO EXK table associates the same exchange key to the fact that testing is performed by an ISDN TOS.
4. On a second ODB DATA ROUTING table, the test centers initiating ISDN tests are associated with the ISDN TOS.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DATA ROUTING  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0005
NOTE: NSDB SUBSYSTEM DATA ROUTING TABLE    REL/LEV:              MOD: Y
   FIELD    FIELD
   NAME     VALUE
   ROUTEKEY                *
   NODE1                   *
   NODE2                   *
   NODE3                   *
   NODE4                   *
   NODE5                   *
   NODE6                   *
   NODE7                   *
   NODE8                   *
   NODE9                   *
   NODE10                  *
   LINK_ON   *
```

**Figure 7–12.** ODB DATA ROUTING Table

**Table Key(s)/Table Record Key(s):** INE/none
NMAFAC/State or City and State CLLI
NMASWI/State or City and State CLLI
NMAISDN/State or City and State CLLI
(INQWRK) ITS–ISDN SEC Code/ISDN
(ADDTST) WFA Center/6–digit Exchange Key
WFA/SNDMTH, SNDRSL, SNDSWP, UNSSO

**Table Type:** DSECT

**Initial Issue:** 3.0

**Data Supplied by:** BCC

**Updating Cautions:** None

**Module(s):** VJDBI01, VJDMI03

## 7.9.2 Field Descriptions

The field values used for this table vary for each INTERFACE ID. To locate the appropriate field value description, you need to find the INTERFACE ID for the combination of systems that you are interested in. The usage of the field, the values to be entered, and table key information are described for each INTERFACE ID.

### 7.9.2.1 The ROUTEKEY Field

ROUTEKEY

Full Name: Route Key

Definition: This field serves as the key data element that other elements are mapped to. The ROUTEKEY field does not have a single definition. The value has a different meaning that depends upon the element used as the 'key'.

Data Content: 8 A/N

**INTERFACE ID = CRS-NSDB-WFA/C**

**Usage for telephone number swaps sent to WFA/C;**

The ROUTEKEY value for telephone number swap notification to WFA/C is SNDSWP (send swap).

**INTERFACE ID = SOAC-NSDB-WFA/C**

**Usage for SOAC service order routing to WFA/C;**

The ROUTEKEY value for service order routing to WFA/C should be UNSSO (unsolicited service order order).

The ROUTEKEY value for the match message should be SNDMTH (send match).

**INTERFACE ID = TIRKS-NSDB-NMA/Facilities**
**= TIRKS-NSDB-NMA/Pass Through**
**= TIRKS-NSDB-NMA/Switch**

**Usage for NMA routing;**

The common language name of either a state, or the city and state in combination. The entry here is mapped to the ROUTEBY field in the ODB ROUTEBY table. If the ROUTEKEY entry is a state identifier, it works with the AZST method on the ODB ROUTEBY table. If the ROUTEKEY entry is a city and state combination, it works with the AZCITYST method on the ODB ROUTEBY table.

Sample Entry: NJ (state identified)
PISCNJ (city and state combination identified)

**INTERFACE ID = WFA/C-NSDB-ITS/ISDN**

**Usage for the ITS/ISDN routing;**

There are two ODB DATA ROUTING tables required for the ITS/ISDN routing process. The ROUTEKEY field has a different value in each table.

The first table associates ISDN service with the test centers that initiate the ISDN test requests. For this table, the ROUTEKEY field entry is ISDN.

The second table associates the exchange key that provides ISDN service to the ITS system SECs that provide testing. For this table, the ROUTEKEY field value is comprised of the 6 numerics for the exchange key.

Sample Entry: ISDN (for the first table described above)

908699 (the exchange key value described in the second table above)

### 7.9.2.2 The NODE1 – NODE10 Fields

NODE1

Full Name: Node 1 through Node 10

Definition: These fields are the elements that are associated with the ROUTEKEY field. Like ROUTEKEY, the values entered here describe different things depending upon the table usage.

Data Content: Up to 8 A/N

**INTERFACE ID = CRS-NSDB-WFA/C**

**Usage for telephone number swaps sent to WFA/C;**

The NODE values for the telephone number swap notification to WFA/C are the SEC codes that identify the WFA/C systems.

Sample Entry: VOIT1SEC

**INTERFACE ID = SOAC-NSDB-WFA/C**

**Usage for SOAC service order routing to WFA/C;**

The NODE values for service order routing and the match message routing are the System Entity Codes (SEC) that identify the WFA/C systems.

Sample Entry: VOIT1SEC

**NOTE –** To deploy the OPS/INE interface, a node *must* be populated with the WFA/C System Entity Code (SEC). This node needs to be built whether the provisioning source is TIRKS or SOAC in order for OPS/INE to identify which WFA/C system is to receive OPS/INE data.

**INTERFACE ID = TIRKS-NSDB-NMA/Facilities**
**= TIRKS-NSDB-NMA/Pass Through**
**= TIRKS-NSDB-NMA/Switch**

**Usage for NMA routing;**

Identifies the NMA system(s) that receive TIRKS data for circuits carrying the Route Key locations. The entry in this usage is the SEC for the NMA system(s).

Sample Entry: TCISN021

**INTERFACE ID = TIRKS–NSDB & WFA/C–OPS/INE**

**Usage for OPS/INE routing;**

The NODE values are the SEC codes which identify the OPS/INE system(s) that need to receive circuit data from NSDB.

> **NOTE** – To deploy the OPS/INE interface, a node *must also* be populated with the WFA/C System Entity Code (SEC) for a table key value of "WFA" and a table record key value of "UNSSO". This node needs to be built whether the provisioning source is TIRKS or SOAC in order for OPS/INE to identify which WFA/C system is to receive OPS/INE data.

**INTERFACE ID = WFA/C-NSDB-ITS/ISDN**

**Usage for ITS/ISDN routing;**

There are two ODB DATA ROUTING tables required for the ITS/ISDN routing process. The NODE1 through NODE10 field(s) have a different value in each table.

The first table associates ISDN service with the test centers that initiate the ISDN test requests. For this table, the NODE1 through NODE10 fields are the CLLI codes that identify the test centers.

The second table associates the exchange key that provides ISDN service to the ITS system SECs that provide testing. For this table, the NODE1 through NODE10 fields are the SEC codes that identify the ITS systems that test ISDN circuits for the exchange key in the ROUTEKEY field.

Sample Entry: ATLNGA01SSC (for the first table described above)

TCISN099 (the ITS SEC code value described in the second table above)

### 7.9.2.3 The LINK_ON Field

**INTERFACE ID = CRS–NSDB–WFA/C**

**SOAC–NSDB–WFA/C**

**TIRKS–NSDB–NMA–Facilities**

**TIRKS–NSDB–NMA/Pass Through**

**TIRKS–NSDB–NMA/Switch**

**WFA/C–NSDB–ITS/ISDN**

LINK_ON

Full Name: Link Activation Flag

Definition: This field is used to activate or disable data routing to an external system as defined in the ODB DATA ROUTING table.

Data Content: 1 A or blank

Valid Entries: Y (Yes, the link is active)

N (No, the link is disabled)

Blank (defaults to Yes, the link is active)

### 7.9.3 Creating the Table

**INTERFACE ID = CRS-NSDB-WFA/C**

The ODB DATA ROUTING table is used to route the Telephone Number swap information, initiated by the Centrex Rearrangement System (CRS), to WFA/C.

Follow the steps below to create this table.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB DATA ROUTING in the TABLE NAME field and press the PF1 function key to FIND.
3. Type WFA in the TABLE KEY field.
4. On the returned screen, type SNDSWP in the ROUTEKEY field.
5. Type in the SEC codes that correspond to the WFA systems which will be receiving telephone number swap messages in the NODE1 through NODE10 fields.
6. Press the PF4 function key to ADD the values to the table.
7. Verify the table is complete by performing a FIND. To do this, the TABLE NAME field must be ODB DATA ROUTING, the TABLE KEY field must be WFA, and the TABLE RECORD KEY must be SNDSWP. Press the PF1 function key to see the completed table.

**INTERFACE ID = SOAC-NSDB-WFA/C**

Two versions of this table need to be created for this interface. The first version is used to route service order messages to WFA/C. The second version is used to route a match message to WFA/C. The match message is the result of the combining of SOAC data and TIRKS system data from a service order.

For the first version, the NSDB System Administrator must identify the WFA/C systems that need to receive SOAC service orders. The TABLE KEY used is WFA, the ROUTEKEY is UNSSO, and each WFA/C system is uniquely identified by its SEC in the NODE1 through NODE10 fields.

The second version (for routing the match message) has a TABLE KEY of WFA, a ROUTEKEY of SNDMTH, and the NODE1 through NODE10 fields contain the SEC codes for the WFA/C systems.

CREATION OF THE ODB DATA ROUTING TABLE

**To create the first version of this table for service order data routing to WFA/C, follow the steps below:**

1. From a cleared screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB DATA ROUTING in the TABLE NAME field, and press the PF1 function key to FIND.
3. Type WFA in the TABLE KEY field.
4. On the returned screen, type UNSSO in the ROUTEKEY field.
5. Type in the SEC codes that correspond to the WFA systems which will be receiving service orders in the NODE1 through NODE10 fields.

   To deploy the OPS/INE interface, a node *must* be populated with the WFA/C System Entity Code (SEC). This node needs to be built whether the provisioning source is TIRKS or SOAC in order for OPS/INE to identify which WFA/C system is to receive OPS/INE data.
6. Press the PF4 function key to ADD the values to the table.
7. Verify that the table is complete by performing a FIND. To do this, the TABLE NAME field must be ODB DATA ROUTING, the TABLE KEY field must be WFA, and the TABLE RECORD KEY must be UNSSO. Press the PF1 function key to see the completed table.

**To create the second version of this table for match message routing to WFA/C, follow the steps below:**

1. From a cleared screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB DATA ROUTING in the TABLE NAME field and press the PF1 function key to FIND.
3. Type WFA in the TABLE KEY field.
4. On the returned screen, type SNDMTH in the ROUTEKEY field.
5. Type in the SEC codes that correspond to the WFA systems which will be receiving match messages in the NODE1 through NODE10 fields.
6. Press the PF4 function key to ADD the values to the table.
7. Verify that the table is complete by performing a FIND. To do this, the TABLE NAME field must be ODB DATA ROUTING, the TABLE KEY field must be WFA, and the TABLE RECORD KEY must be SNDMTH. Press the PF1 function key to see the completed table.

CREATION OF THE ODB DATA ROUTING TABLE

**INTERFACE ID = TIRKS-NSDB-NMA/Facilities**
**= TIRKS-NSDB-NMA/Pass Through**
**= TIRKS-NSDB-NMA/Switch**

The creation of the ODB DATA ROUTING table must be done with the distribution strategy for the NMA system in mind. This table works in connection with the ODB ROUTEBY table. The number of required ODB DATA ROUTING tables needed for successful distribution of TIRKS data to NMA depends upon the number of NMA systems deployed, the NMA interfaces in use, and the desired routing methods used. Each BCC has unique characteristics that can influence the number of ODB DATA ROUTING tables required.

The procedure for creating the ODB DATA ROUTING table is described for two situations. The first situation is used when the System Administrator needs to associate multiple NMA systems to a single location (either state, or city and state in combination). The second situation is used when a single NMA system needs to be associated with multiple locations (either states, or city and state combinations).

**To create the ODB DATA ROUTING table for a single region served by multiple NMA systems, follow the steps below:**

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB DATA ROUTING in the TABLE NAME field, and press the PF1 function key to FIND.
3. On the returned screen, type the NMA interface identifier in the TABLE KEY field. (This is the same value used in the SYSID field in the ODB ROUTEBY table. Valid entries are NMAFAC or NMASWI.)
4. Type the location served by multiple NMA systems in the ROUTEKEY field. (This is the common language value for either a state, or a city and state combination.)
5. Type in the SEC codes that correspond to the multiple NMA systems in the NODE1 through NODE10 fields.
6. Press the PF4 function key to ADD the values to the table.
7. Verify that the table is complete by performing a FIND. To do this, the TABLE NAME field must be ODB DATA ROUTING, the TABLE KEY field must be the value selected in step 3 above (NMAFAC or NMASWI), and the TABLE RECORD KEY must be the location selected in step 4 above. Press PF1 to see the completed table.

## CREATION OF THE ODB DATA ROUTING TABLE

**To create the ODB DATA ROUTING table for a single NMA system serving multiple regions, follow the steps below:**

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB DATA ROUTING in the TABLE NAME field, and press the PF1 function key to FIND.
3. On the returned screen, type the NMA interface identifier in the TABLE KEY field. (This is the same value used in the SYSID field in the ODB ROUTEBY table. Valid entries are NMAFAC or NMASWI.)
4. Type the first location served by the NMA system in the ROUTEKEY field. (This is the common language value for either a state, or a city and state combination.)
5. Type the SEC code that represents the NMA system in the NODE1 field.
6. Press the PF4 function key to ADD the values to the table.
7. To add the second location served by the NMA system, overtype the value in the ROUTEKEY field with the second location value. When this is done, press the PF4 function key to ADD the record.
8. Continue adding locations in this manner until all desired locations are recorded.
9. Verify that the tables are complete by performing a SCAN COMMAND. To do this, the TABLE NAME field must be ODB DATA ROUTING, and the TABLE KEY field must be the value selected in step 3 above (NMAFAC or NMASWI). Type SCAN in the COMMAND field and press the PF1 function key. All ROUTEKEY values will be displayed.

CREATION OF THE ODB DATA ROUTING TABLE

**INTERFACE ID = TIRKS–NSDB & WFA/C–OPS/INE**

The ODB DATA ROUTING table is used to route data from NSDB for multi–point and point–to–point circuits to the necessary OPS/INE system(s).

Follow the steps below to create this table:

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB DATA ROUTING in the TABLE NAME field and press the PF1 function key to FIND.
3. Type INE in the TABLE KEY field.

   To deploy the OPS/INE interface, a node *must also* be populated with the WFA/C System Entity Code (SEC) for a table key value of "WFA" and a table record key value of "UNSSO". This node needs to be built whether the provisioning source is TIRKS or SOAC in order for OPS/INE to identify which WFA/C system is to receive OPS/INE data.
4. On the returned screen, type the locations supported by OPS/INE (i.e., city/state or state) in the ROUTEKEY field. (See also the ODB ROUTEBY TTS table.)
5. Type in the SEC codes that correspond to the specified OPS/INE system(s) in the NODE1 through NODE10 fields.
6. Press the PF4 function key to ADD the values to the table.
7. Verify the table is complete by performing a FIND. To do this, the TABLE NAME field must be ODB DATA ROUTING, the TABLE KEY field must be INE, and the TABLE RECORD KEY must be left blank. Press the PF1 function key to see the completed table.

CREATION OF THE ODB DATA ROUTING TABLE

**INTERFACE ID = WFA/C-NSDB-ITS/ISDN**

The NSDB System Administrator must create two versions of the ODB DATA ROUTING table to support the ITS/ISDN interface. The first table is used to associate ISDN service to the test centers that initiate ISDN test requests. The second version associates an exchange key that provides ISDN service to the ITS systems which will perform the testing. These two tables are used in conjunction with the ODM EXK TO SW ID and ODM TOS TO EXK tables to provide ITS/ISDN data routing for NSDB.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB DATA ROUTING in the TABLE NAME field, and press the PF1 function key to FIND.
3. On the returned screen, type the ITS SEC code for the ITS system that tests ISDN in the TABLE KEY field.
4. Type ISDN in the ROUTEKEY field.
5. Type in the CLLI codes that correspond to the test centers which will be sending ISDN test requests to the ITS system in the TABLE NAME field in the NODE1 through NODE10 fields.
6. Press the PF4 function key to ADD the values to the table.
7. Verify that the table is complete by performing a FIND. To do this, the TABLE NAME field must be ODB DATA ROUTING, the TABLE KEY field must be the value selected in step 3 above, and the TABLE RECORD KEY must be ISDN. Press the PF1 function key to see the completed table.

### 7.9.4 Maintaining the Table

The ODB DATA ROUTING table controls the routing of data for several interfaces. When the destination of data changes for these interfaces, the table may require modification. Additional changes may be needed for the NMA interfaces that use routing schemes identified in the ODB ROUTEBY table. (See the ODB ROUTEBY table section for routing scheme descriptions.)

The ODB DATA ROUTING table should be reviewed and modified as necessary when new interfaces are turned up, or when existing interfaces need data routed to new or different locations.

## 7.10 ODB DATAPORT

### 7.10.1 Table Description

This table is used to determine which bank type codes have dataport capacity for OPS/INE operations.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DATAPORT      TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: BANK CODE RETURNS FUNCTION/HECIG CODES   REL/LEV:            MOD: Y
   FIELD     FIELD
   NAME      VALUE
   BANKCODE        *
   FUNCOD01              *
   FUNCOD02              *
   FUNCOD03              *
   FUNCOD04              *
   FUNCOD05              *
   FUNCOD06              *
   FUNCOD07              *
   FUNCOD14              *
   FUNCOD09              *
   FUNCOD10              *
   FUNCOD11              *
   FUNCOD12              *
   FUNCOD13              *
   FUNCOD14              *
```

**Figure 7–13.** ODB DATAPORT Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DATAPORT      TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: BANK CODE RETURNS FUNCTION/HECIG CODES   REL/LEV:            MOD:
   FIELD    FIELD
   NAME     VALUE
   HECIG01          *
   HECIG02          *
   HECIG03          *
   HECIG04          *
   HECIG05          *
```

**Figure 7–14.** ODB DATAPORT Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | INE |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | Bellcore |
| **Updating Cautions:** | Cross–check this table with the ODB BANKCODES table. |
| **Related Table(s):** | ODB BANKCODES |
| **Format(s):** | None |
| **Module(s):** | VJDBICK |

### 7.10.2 Field Descriptions

BANKCODE

Full Name: Bank Type Code

Definition: The code that identifies the network elements that can be provisioned by the OPS/INE system for a TIRKS–generated circuit.

Data Content: 6 A/N

Valid Entries: Any valid bank type code

FUNCOD01 through FUNCOD14

Full Name: Function Code

Definition: The code used in the provisioning system to locate the equipment to be assigned to the circuit.

Data Content: 8 A/N

Valid Entries: Any valid function code

HECIG01 through HECIG05

Full Name: Human Equipment Catalog Item Group Code

Definition: The code that identifies a group of equipment items and maps to an OPS/INE device code such as T1, T1C, SRDM or T1DM.

Data Content: 8 A/N

Valid Entries: Any valid HECIG code

### 7.10.3 Creating the Table

This table is prepopulated at Bellcore. To view data on this table, follow these steps:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB DATAPORT" in the TABLE NAME field.
3. Type "INE" in the TABLE KEY field.
4. Perform a FIND by pressing the PF1 function key.

**NOTE** – The information in this table is equivalent to that in the INE DATAPORT TTS table presently stored in the TIRKS system.

### 7.10.4 Maintaining the Table

The ODB DATAPORT table is prepopulated with data, and is maintained by Bellcore.

## 7.11 ODB DEBUG GLOBAL

### 7.11.1 Table Description

This table is used to support:

- dynamic file allocation options (these can be set by using the "DEFAULT" table key or can be tailored to a specific IMS user by creating a table for that IMS user ID).
- debugging for all IMS messages that contain a specific IMS user ID [valid only for NSDB Message Processing Program (MPP) transactions that do not support a screen].

> **NOTE** – Unless you are very familiar with ptrace and debugging processes, don't use the ODB DEBUG GLOBAL or ODB DEBUG TRANS tables without Bellcore's assistance. For advice, call Bellcore's Technical Support Center (TSC) at 908–699–7900.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DEBUG GLOBAL  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0008
NOTE: ODB GLOBAL DEBUG DATA                      REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   GLBUSRID          *
   GLBPTRCO  *
   GLBDBGO   *
   GLBPTRC                                                        *
   GLBDGVAL  *
   DSNAME                                                *
   APDUSRID  *
   PRINTER           *
   DISKUNIT          *
   VIOUNIT           *
   LRECL        *
   BLKSIZE        *
   GLBDATA                                                             *
```

**Figure 7–15.** ODB DEBUG GLOBAL Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | DEFAULT or IMS user ID |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | Check the second debug table, ODB DEBUG TRANS. |
| **Related Table(s):** | ODB DEBUG TRANS |
| **Format(s):** | All WFA/NSDB formats |
| **Module(s):** | VJDBDBG |

### 7.11.2 Field Descriptions

GLBUSRID

Full Name: Global User ID

Definition: This field allows debugging by IMS user ID for all NSDB Message Processing Program (MPP) transactions that do not support a screen.

- The value for this field must match the IMS user ID (which must not be blank) from the IO PCB in order to facilitate debugging when either the GLBPTRCO field or the GLBDBGO field is set to "Y".
- If the TTS value for this field is "DEFAULT", it is used only to obtain the DISKUNIT, VIOUNIT, LRECL, BLKSIZE, and DSNAME when they are not available from other TTS debug tables or from user screen input. Debugging *cannot* be turned on from this TTS table occurrence; the global ptrace and debug fields (e.g., GLBPTRCO and GLBDGVAL) are ignored.

Data Content: 8 A/N

Valid Entries: Valid IMS user IDs or DEFAULT

GLBPTRCO

Full Name: Global Ptrace On

Definition: A switch to show whether a global ptrace should be performed. This switch allows the user to leave data in the GLBPTRC field without turning ptrace on.

Data Content: 1 A

Valid Entries: Y (Yes, turn ptrace on)

N or blank (No, do *not* turn ptrace on; N is the default)

GLBDBGO

Full Name: Global Debug On

Definition: A switch to show whether a global debug should be performed. This switch allows the user to leave data in the DSNAME field without turning debug on.

Data Content: 1 A

Valid Entries: Y (Yes, turn debug on)

N or blank (No, do *not* turn debugging on; N is the default)

GLBPTRC

Full Name: Global Ptrace Information

Definition: Data to be passed to the ptrace modules when the GLBPTRCO field contains a "Y" (Yes).

Data Content: 55 A/N

Example: PT DL –P=TKODB15

GLBDGVAL

Full Name: Global Debug Value

Definition: Defines the level of debugging to be performed.

Data Content: 1 A/N

Valid Entries: 0 through 9 (9 is the highest level and provides the most detail)

? ( "?" tells the debug module to open the dataset, but set the debug value to 0. It can be used to detect print statements that do not test for a debug value.)

DSNAME

Full Name: Dataset Name

Definition: The dataset name defining the name of the debug dataset to be allocated. The transaction name will be appended to the dataset name found in this field.

Data Content: 44 A/N

Example: TKODB15.$DEBUG

APDUSRID

Full Name: Append User ID

Definition: Indicates whether the user ID should also be appended to the debug dataset name to be allocated.

Data Content: 1 A

Valid Entries: Y (Yes, append the user ID to the debug dataset)

N or blank (No, do *not* append the user ID to the debug dataset; N is the default)

PRINTER

Full Name: Printer

Definition: The location to which LTERM debug output should be sent. If the DSNAME field is populated, the value in this field is ignored.

Data Content: 8 A/N

Valid Entries: Any valid LTERM

DISKUNIT

Full Name: Disk (DASD) Unit Name

Definition: The unit type or parameter for the debug dataset when debug writes to a disk.

Data Content: up to 8 A/N

Valid Entries: A valid DASD device type or group name (contact ISO). "DISK" is the default.

VIOUNIT

Full Name: Esoteric Alias Virtual Unit Name

Definition: The unit type for the debug dataset when debug writes to a virtual dataset. This field is used when the Esoteric Alias Virtual Unit name is not the literal "VIO".

Data Content: up to 8 A/N

Valid Entries: "VIO" is the default. This field is blank if the Esoteric Alias Virtual Unit name equals "VIO". Otherwise, it is populated with the name defined in the MVS GEN (contact the ISO organization for the alias name).

LRECL

Full Name: Logical Record Length

Definition: The logical record length desired for the debug dataset.

Data Content: 3 N

Valid Entries: Any numeric combination.
"81" is the default.

BLKSIZE

Full Name: Block Size

Definition: Defines the block size for the debug dataset.

Data Content: 5 N

Valid Entries: Any numeric combination.
"00000" is the default.

**NOTE –** If SMS is not being used, the block size must equal or exceed the LRECL.

Examples: 00000 (if DASD is managed by SMS, an IBM product)
09600
27998

GLBDATA

| | |
|---|---|
| Full Name: | Global Data |
| Definition: | This field is used to avoid changes to the ODB DEBUG GLOBAL TTS table (?) during minor releases. The only tag allowed in this field is "@@PT@@", which tells the debug module to write its own debug data to ptrace. When "@@PT@@" is found, any ptrace data in the other ptrace fields is ignored. |
| Data Content: | 59 A/N |
| Valid Entries: | @@PT@@ |
| Examples: | SYSDA, DASD, 3390, or 3380 |

### 7.11.3 Debug Keywords Supported on Screen Message Line

The following debugging keywords are still supported for the NSDB screen message (bottom) line:

| | |
|---|---|
| BLK=xxxxx | The "BLK=" keyword indicates the block size to be used when allocating the debug dataset. "xxxxx" is the block size. |
| DEBUGx | The keyword "DEBUG" indicates that debugging is desired. The "x" specifies the level of debugging desired. It can be a value from 0 through 9 or the value "?", which sets the value to "0"and tells the debugging modules to open SYSPRINT. "?" can be used to avoid a test for the debug value but test for the modules that write to SYSPRINT. *This is a required keyword.* |
| DG=xxxxxxxx | The "DG=" keyword indocates the name of the dataset to be allocated. "xxxxxxxx" is the dataset name which follows MVS dataset naming conventions. *This is a required keyword.* |
| DU=xxxxxxxx | The "DU=" keyword indicates the unit type to be used for a disk dataset. "xxxxxxxx" is the unit type. |
| LRCL=xxx | The "LRCL=" keyword indicates the logical record length to use when allocating the debug dataset. "xxx" is the logical record length. |
| LT=xxxxxxxx | The "LT=" keyword indicates that debug output should be sent to the LTERM specified as "xxxxxxxx". |
| NOBLOCK | The "NOBLOCK" keyword indicates that the SYSPRINT should have a blocking factor of one record per block. |
| NOSPIE | The "NOSPIE" keyword indicates that SPIE should be turned off. |
| PLIDUMP | The "PLIDUMP" keyword indicates that a dump should be done when an application error is encountered. |
| VU=xxxxxxxx | The "VU=" keyword indicates the unit type to be used for a virtual I/O dataset. "xxxxxxxx" is the unit type. |
| \|?XAID options | The "\|?XAID" keyword indicates that a ptrace is desired. "options" are the ptrace options desired. Only ptrace options can follow the "\|?XAID" keyword. |
| @@PT@@ | Turn on ptrace for the debug module. |

### 7.11.4 General Information about the Debugging Tables

Debugging will not take place if the DEBUG_ON option of the ODB CO OPTIONS TTS table is set to "N".

If the DEBUG_ON flag in the ODB CO OPTIONS TTS table is set to "Y", the system will check the ODB DEBUG GLOBAL table for a valid GLBUSRID. If a valid GLBUSRID is found and either the GLBPTRCO option or the GLBDBGO option (or both) are set to "N", the system *will not* check the ODB DEBUG TRANS table for transaction–level information.

If, on the other hand, a valid GLBUSRID is found and the GLBPTRCO and GLBDBGO options are set to "Y", the system *will* check the ODB DEBUG TRANS table, and every transaction associated with the run will be debugged.

If you wish to initiate debugging for a single transaction that has been defined in the ODB DEBUG TRANS table, you must *not* allocate the ODB DEBUG GLOBAL table with your user ID. This will cause the system to use "DEFAULT" as the user ID when allocating the dataset.

### Example: Debugging All NSDB Transactions for IMS User ID "TKODB15"

The following steps illustrate an example which turns on ptrace and debugging for all transactions initiated by the IMS user ID "TKODB15":

> **NOTE** – Yes ("Y") ptrace and debug values by IMS user ID in ODB DEBUG GLOBAL table will override transaction–specific ptrace and debug options that are set in the ODB DEBUG TRANS table.

> **NOTE** – This process *will not* debug WFA/C transactions that update NSDB databases.

[1] From a cleared screen, type "/FOR VJTDSO" and press the ENTER key.

[2] Type "ODB DEBUG GLOBAL" in the TABLE NAME field, type "DEFAULT" in the TABLE RECORD KEY field, and press the FIND PF1 function key.

[3] Change the value in the GLBUSRID field to "TKODB15".

[4] In the GLBPTRCO field, type a "Y".

[5] Enter ptrace data in the GLBPTRC field. In this case, ptrace data is "PT DL –P=TKODB15".

[6] Set the GLBDBGO field to "Y", and set the debug level in field GLBDGVAL to a "9".

[7] Make sure that the DSNAME field has valid dataset qualifiers. In this case, the dataset qualifiers are "TKODB15.$DEBUG".

[8] ADD the new ODB DEBUG GLOBAL TTS entry for IMS user ID "TKODB15" by pressing the PF4 function key.

[9] Use the FIND PF1 function key to obtain the ODB CO OPTIONS TTS table.

[10] Set the DEBUG_ON field to "Y".

[11] Update the ODB CO OPTIONS table by pressing the PF5 function key.

Ptrace and debugging is turned on for all transactions associated with IMS user ID TKODB15.

> **NOTE** – To *turn off* global ptrace and debugging processes, change the values in the GLBPTRCO and GLBDBGO fields on the ODB DEBUG GLOBAL table to "N", and change the DEBUG_ON option in the ODB CO OPTIONS table to "N".

> **NOTE** – Refer to the next section on the ODB DEBUG TRANS table to view samples of populated tables and instructions for debugging a single transaction.

## 7.12 ODB DEBUG TRANS

### 7.12.1 Table Description

This table is used to support selective debugging and/or ptrace by transaction. Note that the transaction–level ptrace and debug options in this table can be overridden by Y (Yes) options in the ODB DEBUG GLOBAL TTS table.

> **NOTE –** Unless you are very familiar with ptrace and debugging processes, don't use the ODB DEBUG GLOBAL or ODB DEBUG TRANS tables without Bellcore's assistance. For advice, call Bellcore's Technical Support Center (TSC) at 908–699–7900.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DEBUG TRANS   TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                              # OF RECORDS: 0002
NOTE: DEBUG OPTIONS PER TRANSACTION           REL/LEV:            MOD: Y
   FIELD     FIELD
   NAME      VALUE
   LCLTRAN           *
   LCLPTRCO  *
   LCLDBGO   *
   LCLPTRC                                                        *
   LCLDGVAL  *
   NOBLOCK   *
   PLIDUMP   *
   NOSPIE    *
   ACNO#                                                *
   ORDNO#                          *
   CKTID#                                               *
   CAC#              *
   TRK#                   *
   INT_ID#             *
   DBGSTART            *
```

**Figure 7–16.** ODB DEBUG TRANS Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DEBUG TRANS   TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0002
NOTE: DEBUG OPTIONS PER TRANSACTION              REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   DBGEND              *
   DEBUGKEY                                                        *
```

**Figure 7–17.** ODB DEBUG TRANS Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Transaction Name (e.g., VJDBL4T) |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | Check the first table, ODB DEBUG GLOBAL. The dynamic allocation data should be used with the table record key "DEFAULT" or a private user ID. |
| **Related Table(s):** | ODB DEBUG GLOBAL |
| **Format(s):** | None |
| **Module(s):** | VJDBDBG |

### 7.12.2 Field Descriptions

LCLTRAN

Full Name: Local Transaction

Definition: Designates the NSDB Message Processing Program (MPP) transaction to which this debug applies.

Data Content: 8 A/N

Example: VJDBL4T

LCLPTRCO

Full Name: Local Ptrace On

Definition: A switch to show whether a ptrace should be performed for the transaction identified in the LCLTRAN field. This switch allows the user to leave data in the LCLPTRC field without turning ptrace on.

Data Content: 1 A

Valid Entries: Y (Yes, turn ptrace on)

N or blank (No, do *not* turn ptrace on; N is the default)

LCLDBGO

Full Name: Local Debug On

Definition: A switch to show whether a debug should be performed for the transaction identified in the LCLTRAN field. This switch allows the user to leave data in the DSNAME field without turning debug on.

Data Content: 1 A

Valid Entries: Y (Yes, turn debug on)

N or blank (No, do *not* turn debugging on; N is the default)

LCLPTRC

Full Name: Local Ptrace Information

Definition: Data to be passed to the ptrace modules when the LCLPTRCO field contains a "Y" (Yes).

Data Content: 55 A/N

Example: PT DL –P=VJDBL4T

LCLDGVAL

Full Name: Local Debug Value

Definition: Defines the level of debugging to be performed.

Data Content: 1 A/N

Valid Entries: 0 through 9 (9 is the highest level and provides the most detail)

? ( "?" tells the debug module to open the dataset, but set the debug value to 0. It can be used to detect print statements that do not test for a debug value.)

NOBLOCK

Full Name: No Block Feature

Definition: Sets the blocking factor to one logical record per block. This feature is useful for 0C4 transaction errors.

Data Content: 1 A

Valid Entries: Y (Yes, turn on the NOBLOCK feature)

N or blank (No, do *not* turn on the NOBLOCK feature; N is the default)

PLIDUMP

Full Name: PLI Dump

Definition: Indicates that a dump should be done when an application error is encountered.

Data Content: 1 A

Valid Entries: Y (Yes, perform a dump when an application error is encountered)

N or blank (No, do *not* perform a dump when an application error is encountered; N is the default)

NOSPIE

Full Name: No SPIE

Definition: Flag to turn off SPIE

Data Content: 1 A

Valid Entries: Y (Yes, turn off SPIE)

N or blank (No, leave SPIE on; N is the default)

ACNO#

Full Name: Activity Number

Definition: When this field is populated, it indicates a desire to perform a debug or ptrace for messages with the designated activity number. The value in this field is ignored if an ACNO is not passed to the debug module by the calling program.

Data Content: 46 A/N

ORDNO#

Full Name: Order Number

Definition: When this field is populated, it indicates a desire to perform a debug or ptrace for messages with the designated order number. The value in this field is ignored if an ORDNO is not passed to the debug module by the calling program.

Data Content: 20 A/N

CKTID#

Full Name: Circuit ID Number

Definition: When this field is populated, it indicates a desire to perform a debug or ptrace for messages with the designated circuit ID number. The value in this field is ignored if a CKTID is not passed to the debug module by the calling program.

Data Content: 46 A/N

CAC#

Full Name: Circuit Access Code Number

Definition: When this field is populated, it indicates a desire to perform a debug or ptrace for messages with the designated circuit access code number. The value in this field is ignored if a CAC is not passed to the debug module by the calling program.

Data Content: 8 A/N

TRK#

Full Name: Tracking Key Number

Definition: When this field is populated, it indicates a desire to perform a debug or ptrace for messages with the designated tracking key or circuit layout number. The value in this field is ignored if a TRK or CLO is not passed to the debug module by the calling program.

Data Content: 13 A/N

INT_ID#

Full Name: Internal ID Number

Definition: When this field is populated, it indicates a desire to perform a debug or ptrace for messages with the designated internal ID number. The value in this field is ignored if an internal ID is not passed to the debug module by the calling program.

Data Content: 10 N

DBGSTART

Full Name: Debug Start

Definition: This field is used in conjunction with the DBGEND field to establish a range of circuits for debugging that is transaction or application–oriented. It tells the calling module which circuits in a multiple circuit message should be debugged. (A work order may contain many circuit updates, but not all of the circuit updates will need to be debugged.)

Data Content: 10 N (Bellcore's recommendation is to limit the specified range to less than 100 circuits)

DBGEND

Full Name: Debug End

Definition: This field is used in conjunction with the DBGSTART field to establish a range of circuits for debugging that is transaction or application–oriented. It tells the calling module which circuits in a multiple circuit message should be debugged. (A work order may contain many circuit updates, but not all of the circuit updates will need to be debugged.)

Data Content: 10 N (Bellcore's recommendation is to limit the specified range to less than 100 circuits)

DEBUGKEY

Full Name: Debug Key

Definition: This field is reserved for future use.

Data Content: 55 A/N

Valid Entries: not in use

### 7.12.3 Debug Keywords Supported on Screen Message Line

The following debugging keywords are still supported for the NSDB screen message (bottom) line:

| | |
|---|---|
| BLK=xxxxx | The "BLK=" keyword indicates the block size to be used when allocating the debug dataset. "xxxxx" is the block size. |
| DEBUGx | The keyword "DEBUG" indicates that debugging is desired. The "x" specifies the level of debugging desired. It can be a value from 0 through 9 or the value "?", which sets the value to "0"and tells the debugging modules to open SYSPRINT. "?" can be used to avoid a test for the debug value but test for the modules that write to SYSPRINT. *This is a required keyword.* |
| DG=xxxxxxxx | The "DG=" keyword indocates the name of the dataset to be allocated. "xxxxxxxx" is the dataset name which follows MVS dataset naming conventions. *This is a required keyword.* |
| DU=xxxxxxxx | The "DU=" keyword indicates the unit type to be used for a disk dataset. "xxxxxxxx" is the unit type. |
| LRCL=xxx | The "LRCL=" keyword indicates the logical record length to use when allocating the debug dataset. "xxx" is the logical record length. |
| LT=xxxxxxxx | The "LT=" keyword indicates that debug output should be sent to the LTERM specified as "xxxxxxxx". |
| NOBLOCK | The "NOBLOCK" keyword indicates that the SYSPRINT should have a blocking factor of one record per block. |
| NOSPIE | The "NOSPIE" keyword indicates that SPIE should be turned off. |
| PLIDUMP | The "PLIDUMP" keyword indicates that a dump should be done when an application error is encountered. |
| VU=xxxxxxxx | The "VU=" keyword indicates the unit type to be used for a virtual I/O dataset. "xxxxxxxx" is the unit type. |
| \|?XAID options | The "\|?XAID" keyword indicates that a ptrace is desired. "options" are the ptrace options desired. Only ptrace options can follow the "\|?XAID" keyword. |
| @@PT@@ | Turn on ptrace for the debug module. |

### 7.12.4 General Information about the Debugging Tables

Debugging will not take place if the DEBUG_ON option of the ODB CO OPTIONS TTS table is set to "N".

If the DEBUG_ON flag in the ODB CO OPTIONS TTS table is set to "Y", the system will check the ODB DEBUG GLOBAL table for a valid GLBUSRID. If a valid GLBUSRID is found and either the GLBPTRCO option or the GLBDBGO option (or both) are set to "N", the system *will not* check the ODB DEBUG TRANS table for transaction–level information.

If, on the other hand, a valid GLBUSRID is found and the GLBPTRCO and GLBDBGO options are set to "Y", the system *will* check the ODB DEBUG TRANS table, and every transaction associated with the run will be debugged.

If you wish to initiate debugging for a single transaction that has been defined in the ODB DEBUG TRANS table, you must *not* allocate the ODB DEBUG GLOBAL table with your user ID. This will cause the system to use "DEFAULT" as the user ID when allocating the dataset.

## Example: Debugging Transaction VJDBL4T for Order Number NJHN01092

The following steps illustrate an example which turns on ptrace and debugging for transaction VJDBL4T when the Order Number is set to NJHN01092:

[1] From a cleared screen, type "/FOR VJTDSO" and press the ENTER key.

[2] Type "ODB DEBUG GLOBAL" in the TABLE NAME field, type "DEFAULT" in the TABLE RECORD KEY field, and press the FIND PF1 function key.

[3] Make sure that the DSNAME field has valid dataset qualifiers. In this case, the dataset qualifiers are "TKODB00.$DEBUG.IT1" (refer to the sample ODB DEBUG GLOBAL figure below).

[4] Set the APDUSRID field to "N" or blank if you do *not* want the IMS user ID appended to the dataset name. In this case, since the APDUSRID is blank and the DSNAME is "TKODB00.$DEBUG.IT1", the dataset name that will be allocated is "TKODB00.$DEBUG.IT1.VJDBL4T".

[5] Make sure that the values in the DISKUNIT, LRECL, and BLKSIZE fields are valid (refer to the sample ODB DEBUG GLOBAL figure below).

[6] Update the ODB DEBUG GLOBAL table by pressing the PF5 function key.

[7] Move to the ODB DEBUG TRANS TTS table by overtyping "ODB DEBUG GLOBAL" with "ODB DEBUG TRANS", typing (in this case) "VJDBL4T" in the TABLE RECORD KEY field, and pressing the FIND PF1 function key.

[8] On the ODB DEBUG TRANS table, type "Y" in the LCLPTRCO field.

[9] Enter ptrace data in the LCLPTRC field. In this case, ptrace data is "PT DL –P=VJDBL4T" (refer to the sample ODB DEBUG TRANS figures below).

[10] Set the LCLDBGO field to "Y", and set the debug level in field LCLDGVAL to a "1" (refer to the sample ODB DEBUG TRANS figures below).

[11] In the ORDNO# field, type the Order Number "NJHN01092".

[12] Update the ODB DEBUG TRANS table by pressing the PF5 function key.

[13] Use the FIND PF1 function key to obtain the ODB CO OPTIONS TTS table.

[14] Set the DEBUG_ON field to "Y".

[15] Update the ODB CO OPTIONS table by pressing the PF5 function key.

Ptrace and debugging is turned on for transaction VJDBl4T in association with Order Number NJHN01092.

> **NOTE –** To *turn off* ptrace and debugging for a single transaction, change the values in the LCLPTRCO and LCLDBGO fields on the ODB DEBUG TRANS table to "N", and change the DEBUG_ON option in the ODB CO OPTIONS table to "N".

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DEBUG TRANS   TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:  VJDBL4T                      # OF RECORDS: 0003
NOTE: DEBUG OPTIONS PER TRANSACTION            REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   DBGEND              *
   DEBUGKEY                                                      *









TTS704I  NEXT COMPLETED
```

**Figure 7–20.** Sample ODB DEBUG TRANS Table (2 of 2)

## 7.13 ODB DEFAULT TAGS

### 7.13.1 Table Description

NSDB performs default processing whenever an ISDN service order is received from SOAC. Once it has been determined that default processing will take place, the system consults the ODM EXK TO SW ID table to determine the ISDN Switch Type. If the ISDN_DEF option is turned on in the ODB CO OPTIONS TTS table, then NSDB will generate the appropriate defaults based on user–defined entries in this table, ODB SDT DEFAULTS, and the ODB DEFAULT TAGS table. Defaults are stored in the Line Record database.

The ODB DEFAULT TAGS table is used to provide a list of candidate tags in support of ISDN Packet services translations when specific parameter values *are not* present on the ISDN service order. (This feature will extend to ISDN Voice services translations in the future.) Some of these tags will be present in the SOAC message, and some will not. If the tags are not present at all, then default values for the tags are provided in the table ODB SDT DEFAULTS. If the tag is present, then additional default parameter settings may need to be added to the tag value sent by SOAC. These additional default parameter values will also be provided in the ODB SDT DEFAULTS table.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB DEFAULT TAGS  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0000
NOTE: ODB EXK/SWITCH DEFAULT TAGS DATA     REL/LEV:            MOD:
   FIELD     FIELD
   NAME      VALUE
   SWITCH              *
   REQ_T1             *
   REQ_T2             *
   REQ_T3             *
   REQ_T4             *
   REQ_T5             *
   REQ_T6             *
   REQ_T7             *
   REQ_T8             *
   REQ_T9             *
   REQ_T10            *
   REQ_T11            *
   REQ_T12            *
   REQ_T13            *
   REQ_T14            *
```

**Figure 7–21.** ODB DEFAULT TAGS Table

| | |
|---|---|
| **Table Key(s):** | P (Packet) or V (Voice) |
| **Table Record Key(s):** | Exchange Key or Switch Type |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | The default values on the ODB SDT DEFAULTS table should be checked whenever data changes in the ODB DEFAULT TAGS table. |
| **Related Table(s):** | ODB SDT DEFAULTS |
| **Format(s):** | None |
| **Module(s):** | VJDBSDT |

### 7.13.2 Field Descriptions

SWITCH

Full Name: Switch

Definition: This field identifies the type of central office switch; the value matches that in the ODM EXK TO SW ID TTS table.

Data Content: 10 A/N

Examples: 5ESS
DMS100

REQ_T1 through REQ_T14

Full Name: Required Tag Names

Definition: The required tag names for a particular service on a particular switch.

Data Content: 8 A

Examples: TWN, PRC, FS, LKLP (sample entries for Packet on a 5ESS switch)
TWN, FS, LCL, FCFZ, LKLP, PCB (sample entries for Packet on a DMS–100 switch)

### 7.13.3 Creating the Table

Follow the steps below to complete the table:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB DEFAULT TAGS" in the TABLE NAME field.
3. Press the FIND PF1 function key.
4. Enter a TABLE KEY of either "P" (Packet) or "V" (Voice – future use), as appropriate.
5. For TABLE RECORD KEY, use either an Exchange Key or Switch Type. (The system will use the Exchange Key first in default processing, and if it is not found, will use the Switch Type.)
6. Type the appropriate tag names in the REQ_T1 through REQ_T14 fields (up to 14 required tag names are accommodated on this table for each switch or exchange key).
7. When all entries are made, ADD the values by pressing the ADD PF4 function key.

At the end of default processing, each of these tags will appear on the Line Record whether or not it was present in the SOAC message. In some cases, additional information will be presented behind the tag.

## 7.14 ODB EXCEPTIONS

### 7.14.1 Table Description

The ODB EXCEPTIONS table is used to identify any ODX Exception Messages that require special handling, and to specify how and where they are processed. Several options are available including the suppression of any or all messages and the selective routing of target messages to specific devices or mailboxes.

(This table is optional and need not be completed if special handling of Error Messages is not required.)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB EXCEPTIONS     TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0006
NOTE: ODB EXCEPTION NOTICE CONTROLS               REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   FROM MSG     *
   TO MSG       *
   SEND FLG  *
   DEST              *
```

**Figure 7–22.** ODB EXCEPTIONS Table

| | |
|---|---|
| **Table Key(s):** | Administrative Area, ALL or MCO |
| **Table Record Key(s):** | From MSG Number |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBPUNT |

### 7.14.2 Field Descriptions

FROM MSG

Full Name: From Message

Definition: A specific exception message or the first number of a range of exception messages. This field identifies a message by its exception notice number.

Data Content: 3 N

Examples: 001, 100, 395

TO MSG

Full Name: To Message

Definition: The last number of a range of exception messages. This field identifies a message by its exception notice number. If only one exception message is desired for special handling, then the value in the FROM MSG field will be the same as the value in the TO MSG field.

Data Content: 3 N

Examples: 001, 100, 395

SEND FLAG

Full Name: Send Flag

Definition: Indicates the action to be taken when the message, or range of messages, is encountered by NSDB.

Data Content: 1 A

Valid Entries: N, P, A, W, or M

N = No message; suppress the exception message
P = Print the message on the IMS printer identified in the DEST field below.
A = Abnormal end (ABEND); for emergency debugging only.
W = Write the message immediately to the user terminal identified in the DEST field below.
M = Mail the message to the user mailbox identified in the DEST field below.

DEST

Full Name: Destination

Definition: The logical terminal (LTERM), printer, or mailbox that receives the selected exception messages.

Data Content: 8 A/N

Valid Entries: PY511CGZ

### 7.14.3 Creating the Table

The NSDB System Administrator can use this optional table to select and control the distribution of exception notices. If special handling of exception notices is not desired, then this table need not be created.

Follow the steps below to complete the table.

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB EXCEPTIONS" in the TABLE NAME field and press the PF1 function key.
3. Type in the desired entries. When complete, ADD the values by pressing the PF4 function key.

NOTE – To find valid table keys, type "scant" in the COMMAND field and press the PF1 function key. Valid table keys include company–defined MCO values, blank, and "ALL". ALL can be used to suppress error messages in the specified range for every MCO.

To find valid table record keys, type a valid table key in the TABLE KEY field, type "scan" in the COMMAND field, and press the PF1 function key. Valid table record keys include all company–defined error messages. Error message values 0 through 99 cannot be used with ALL or MCO table keys, since these values are reserved for company–level messages.

### 7.14.4 Maintaining the Table

When an error occurs in NSDB, three NSDB TTS tables are checked to determine the disposition of the error message. These tables are the ODB EXCEPTIONS, ODB MCO OPTIONS, and the ODB CO OPTIONS tables.

[1] NSDB looks into the ODB EXCEPTIONS table to see if this error message calls for any special processing. This is the table that allows you to reroute or suppress a message. You can reroute or suppress a message for the entire NSDB system, or by individual MCO. (For the TIRKS data that is sent via the EDIIS ISSUWD event, the error message routing is performed using Administrative Area; all other interfaces use MCO.)

[2] If no instructions are found in the ODB EXCEPTIONS table, NSDB looks at the ODB MCO OPTIONS table to see if error routing has been turned on for all messages belonging to an MCO. Here you can specify routing for all messages for an MCO. (For the ISSUWD interface, routing is performed using Administrative Area; all other interfaces use MCO.)

[3] If no routing information is found in the two tables described above, NSDB looks in the ODB CO OPTIONS table for the default error LTERM. This is the location that will receive the error message if no other routing was selected on the first two tables.

Any message that has been suppressed in the ODB EXCEPTIONS table will remain suppressed until the table is updated to stop the suppression. Periodic review of all suppressed messages is recommended so that important notifications are not lost.

## 7.15 ODB EXEC PARM

### 7.15.1 Table Description

The ODB EXEC PARM table is used to tune the parsing and rescheduling process initiated by the VJDBL5T Work Order transaction when it is invoked by TCM for work orders received from SOAC. The ODB EXEC PARM table supports enhancements to process incoming work order messages efficiently, regardless of message size.

Prior to NSDB Release 3.0.3, there was potential for the parsing process to encounter space problems when handling large work order messages. Such space problems have been alleviated by parsing and updating circuits in groups. As each group of circuits is processed, the space allocated for that group is freed for the next group of circuits. The number of circuits per processing group is defined either by the WOCKT# field on the ODB EXEC PARM table or by the default value associated with the transaction VJDBL5T.

In addition, prior to NSDB Release 3.2, the parsing and updating process for large work orders could result in system timeouts. Timeout problems have been eliminated by the development of an automatic rescheduling process for the VJDBL5T transaction. The automatic rescheduling process is triggered by a group counter, which specifies the number of circuit groups that will be processed by the transaction before self-rescheduling. This number is defined either by the WOGRP# field on the ODB EXEC PARM table or by the default value associated with VJDBL5T.

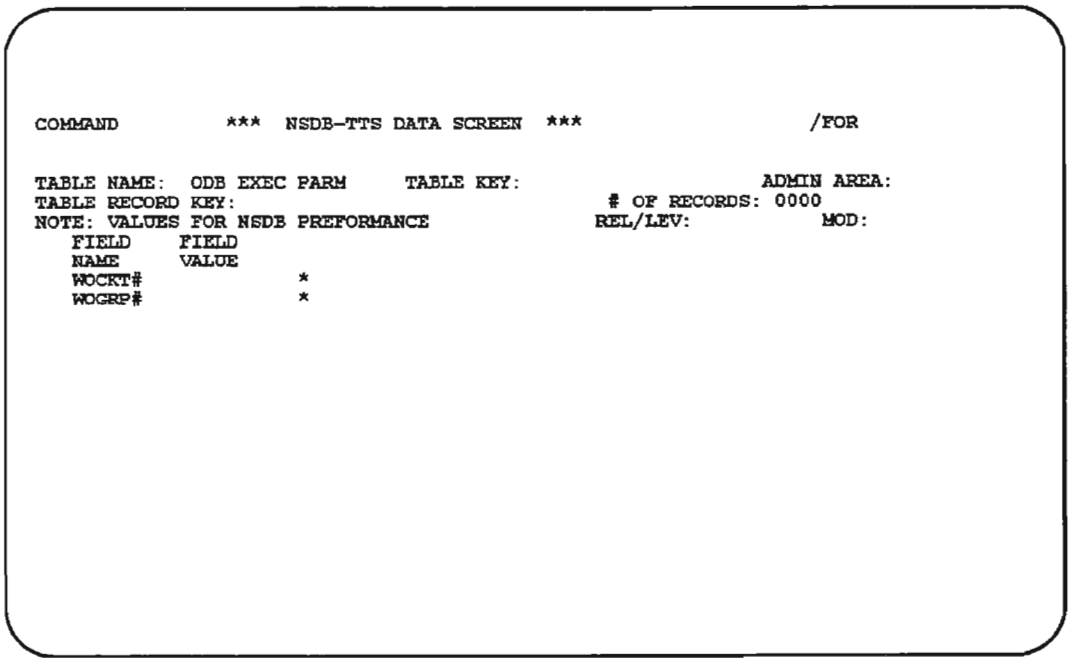



**Figure 7–23.** ODB EXEC PARM Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | Default values are provided in the calling module, VJDBL5P. |
| **Related Table(s):** | None |
| **Format(s):** | None |
| **Module(s):** | VJDBL5P |

### 7.15.2 Field Descriptions

WOCKT#

Full Name: Work Order Circuits – Number of

Definition: The number of circuits in each group to be processed in each invocation of the work order contract.

Data Content: 10 N

Valid Entries: up to 10 numerics
(if blank or zero, the default value from transaction VJDBL5T will be used)

Examples: 40

WOGRP#

Full Name: Work Order Groups – Number of

Definition: The number of circuit groups to be processed before rescheduling the rest of an unparsed TCM message to VJDBL5T.

Data Content: 10 N

Valid Entries: up to 10 numerics
(if blank or zero, the default value from transaction VJDBL5T will be used)

Examples: 8

### 7.15.3 Creating the Table

Follow the steps below to complete the table:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB EXEC PARM" in the TABLE NAME field.
3. Press the FIND PF1 function key.
4. Type the appropriate values in the WOCKT # and WOGRP# fields.
5. When the entries are made, ADD the values by pressing the ADD PF4 function key.

## 7.16 ODB FAC CKT DB

### 7.16.1 Table Description

This table permits the Facility–to–Circuit databases to be loaded separately and in the order that is critical to the circumstances of a given work site.

The ODB FAC CKT DB table is checked during the initial load, during subsequent updates to the database, and whenever a FIND is performed from the WFA/C format OSSFF [refer to the *WFA/C User Manual* (BR 190–513–323) for information about the OSSFF format].

- If you perform a FIND from OSSFF for a facility in one of the databases listed on the ODB FAC CKT DB table and the value for that database is set to "N", the system will return an error message indicating that the database is not installed. A "Y" in the appropriate database field allows the system to attempt a FIND.
- After a database has been loaded, the corresponding field value in the ODB FAC CKT DB table must be set to "Y" or the database will not be updated with new circuit information/updates.

**NOTE** – The CXR_DB does not have to be set to "Y" to perform "upward" FINDS from the OSSFF format since this activity accesses the existing NSDB In–Effect Circuit/Carrier and Message databases.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB FAC CKT DB     TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                # OF RECORDS: 0001
NOTE: ODB FACILITY TO CIRCUIT DATABASE OPTIONS REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   CXR_DB    Y*
   IOCBL_DB  Y*
   LPCBL_DB   *
   SONET_DB   *









TTS210I FIND COMPLETED
```

**Figure 7–24.** ODB FAC CKT DB Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | Once the Facility–to–Circuit tables are populated, the associated database indicator should be set to "Y" (Yes) so the database is automatically updated. |
| **Format(s):** | None |
| **Module(s):** | VJDBIQF, VJDBNO1, VJDBNVM, VJDBUPF |

### 7.16.2 Field Descriptions

CXR_DB

Full Name: Carrier to Circuit Database Flag

Definition: Indicates whether or not the Carrier to Circuit database is loaded.

Data Content: 1 A

Valid Entries: Y (Yes, the Carrier to Circuit database is loaded)

N (No, the Carrier to Circuit database is *not* loaded; N is the default)

IOCBL_DB

Full Name: Interoffice Cable to Circuit Database Flag

Definition: Indicates whether or not the Interoffice Cable to Circuit database is loaded.

Data Content: 1 A

Valid Entries: Y (Yes, the Interoffice Cable to Circuit database is loaded)

N (No, the Interoffice Cable to Circuit database is *not* loaded; N is the default)

LPCBL_DB

Full Name: Loop Cable to Circuit Database Flag

Definition: Indicates whether or not the Loop Cable to Circuit database is loaded. (**NOTE:** this database is not available in the NSDB Release 3.2 time frame.)

Data Content: 1 A

Valid Entries: Y (Yes, the Loop Cable to Circuit database is loaded)

N (No, the Loop Cable to Circuit database is *not* loaded; N is the default)

SONET_DB

Full Name: SONET to Circuit Database Flag

Definition: Indicates whether or not the SONET to SONET and SONET to Circuit database is loaded. (**NOTE:** this database is not available in the NSDB Release 3.2 time frame.)

Data Content: 1 A

Valid Entries: Y (Yes, the SONET to SONET and SONET to Circuit database is loaded)

N (No, the SONET to SONET and SONET to Circuit database is *not* loaded; N is the default)

### 7.16.3 Creating the Table

To specify which Facility–to–Circuit databases are loaded, follow these steps:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB FAC CKT DB" in the TABLE NAME field; leave the TABLE KEY and TABLE RECORD KEY fields blank.
3. Perform a FIND by pressing the PF1 function key.
4. Enter a "Y" or "N" in each field to indicate whether or not each of the four Facility–to–Circuit databases are loaded.
5. Add this information to the ODB FAC CKT DB table by pressing the PF4 function key.

### 7.16.4 Maintaining the Table

The ODB FAC CKT DB table should be updated as individual Facility–to–Circuit databases are loaded.

# 7.17 ODB HECIG TO EQP

## 7.17.1 Table Description

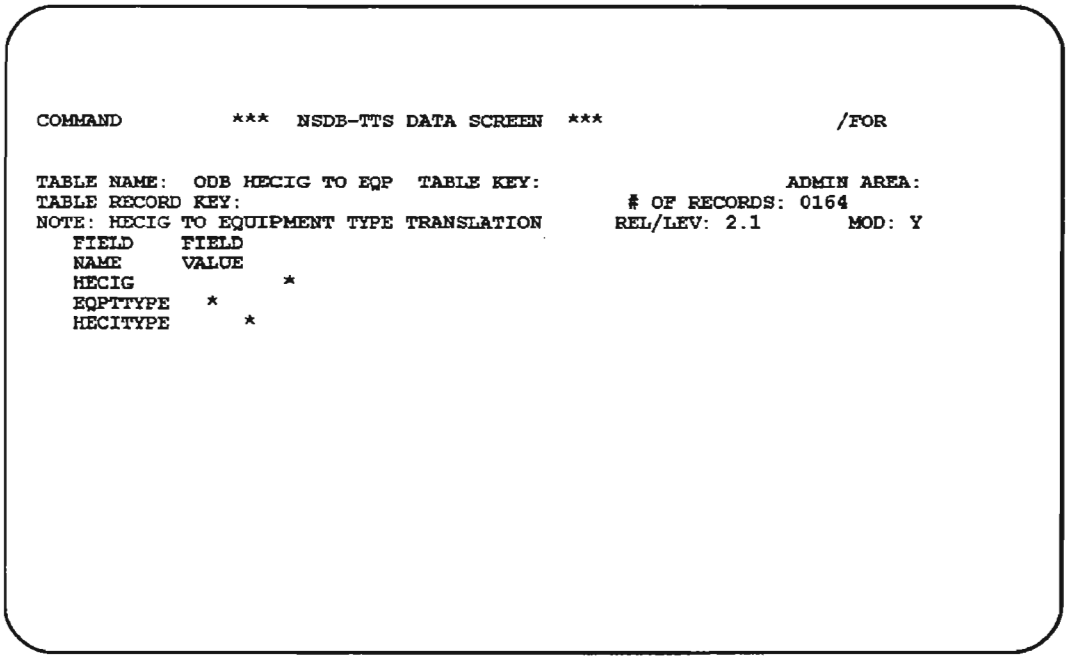
### INTERFACE ID = TIRKS-NSDB-ITS/Digital

The ODB HECIG TO EQP table is used by NSDB for this interface to convert inventoried equipment information, as it is stored by the TIRKS system, to a form that is more suited for processing by NSDB and ITS. The TIRKS system uses the Human Equipment Catalog Item Group (HECIG) types and codes. ITS cannot directly use the equipment information in this convention, so NSDB must translate, store, and pass this information to ITS in a different form. The ODB HECIG TO EQP table provides NSDB with the information needed to convert the TIRKS system HECIG data to an ITS equipment code. The ODB HECIG TO EQP table is used in conjunction with the ODB CO OPTIONS table to build the multi-point views needed to test bridged and point-to-point digital circuits.

Only Digital Data Service (DDS) circuits currently require entries in the ODB HECIG TO EQP table. Entries associate bridge, test point, plug-in, and repeater plug-in equipment with their corresponding HECIG codes and HECIG types. The repeater plug-ins are those commonly used on 56KB DDS circuits.

### INTERFACE ID = TIRKS-NSDB & WFA/C–OPS/INE

The ODB HECIG TO EQP table is used to indicate the equipment supported by OPS/INE. A TABLE KEY of "INE" is required to differentiate OPS/INE from other external systems. This table is prepopulated with data from Bellcore.



```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB HECIG TO EQP  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0164
NOTE: HECIG TO EQUIPMENT TYPE TRANSLATION  REL/LEV: 2.1       MOD: Y
   FIELD    FIELD
   NAME     VALUE
   HECIG            *
   EQPTTYPE   *
   HECITYPE      *
```

**Figure 7–25.** ODB HECIG TO EQP Table

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB HECIG TO EQP  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:  DDMJ400A                  # OF RECORDS: 0164
NOTE: HECIG TO EQUIPMENT TYPE TRANSLATION    REL/LEV: 2.1       MOD: Y
   FIELD     FIELD
   NAME      VALUE
   HECIG     DDMJ400A*
   EQPTTYPE  BR*
   HECITYPE  MJU  *









TTS210I FIND COMPLETED
```

**Figure 7–26.** Populated HECIG TO EQP Table

| | |
|---|---|
| **Table Key(s):** | INE for OPS/INE; None for other interfaces |
| **Table Record Key(s):** | HECIG Code Values |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | This table must be populated before using NSDB. |
| **Module(s):** | VJCDL01, VJDBL1T, VJODU01 |

### 7.17.2 Field Descriptions

HECIG

Full Name: Human Equipment Catalog Item Group Code

Definition: This field identifies the HECIG Codes used in the HECIG to Equipment Type translation.

Data Content: HECIG values consist of a combination of from 1 to 8 alphabetic and numeric characters; no embedded blanks are allowed.

**NOTE:** when identifying OCUs, OCUDPs, or DS1 test points (see below), only the first 7 characters of the HECIG value should be entered.

Valid Entries: Any valid HECIG Code that represents a DDS bridge, a DS1 test access point, plug-in equipment, or a plug-in repeater used in a 56 KB DDS circuit.

For OPS/INE, this code identifies a group of equipment items, and maps to an OPS/INE device code such as T1, T1C, SRDM, or T1DM. HECIG codes have been prepopulated with data from Bellcore for equipment supported by OPS/INE.

EQPTTYPE

Full Name: Equipment Type

Definition: This field is a generic identifier of the HECIG and HECIG types received from the TIRKS system.

Data Content: 2 A

Valid Entries: BR (multi-point bridge equipment)

IN (OPS/INE–supported equipment; Bellcore–populated)

PL (plug-in equipment)

RE (repeater equipment)

RP (repeater plug-in equipment)

TP (test point equipment)

HECIGTYPE

Full Name: HECIG Type

Definition: This field represents a specific type of the HECIG shown above in the HECIG field.

Data Content: 1–5 A/N

Valid Entries:

| | |
|---|---|
| DSO and DS1 | Used when EQPTTYPE field is TP for DDS or DS1 circuits. The edit looks at the HECIG test points on the E (equipment) line of the VJDBCD. |
| IN | Used when EQPTTYPE is IN for equipment supported by OPS/INE. (Bellcore–populated) |
| MJU | Used when EQPTTYPE field is BR and the SUB/MISC field on the WORD document contains the port ID. The edit looks at the HECIG found on the E (equipment) line of the VJDBCD when EQT = EM, EU, or E*. |
| OCU and OCUDP | Used when EQPTTYPE field is PL for OCU edits. The edit looks at the HECIG found on the P (plug) line of the VJDBCD. |
| QMJU | Used when the EQPTTYPE field is BR and the UNIT field on the WORD document contains the port ID. The edit looks at the HECIG found on the E (equipment) line of the VJDBCD when EQT = EM, EU, or E*. |
| 56KB | (used when EQPTTYPE field is RP for 56KB DDS circuits.) |

NOTE: The edit program interprets the HECIGs 'DDTA500' and 'DDTAA001' as test points and does not access this table to see how they are defined. Therefore, no entries for DDTA500 or DDTAA001 are required; this applies to both DS0 and DS1 circuits.

### 7.17.3 Creating the Table

**INTERFACE ID = TIRKS-NSDB-ITS/Digital**

The NSDB System Administrator must create this table with the ODB CO OPTIONS table in mind. These two tables are used to provide accurate mapping of the multi-point DDS circuits that NSDB will encounter. Follow the steps below to create the ODB HECIG TO EQP table:

1. From a cleared screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB HECIG TO EQP in the TABLE NAME field.
3. Perform a FIND by pressing the PF1 function key.
4. On the returned screen, type in the values for the HECIG, EQPTTYPE, and HECITYPE fields that are appropriate for your company.
5. Press the PF4 function key to ADD the values to the table.
6. Repeat this process for each HECIG that NSDB is likely to encounter.

To see a list of HECIG codes in the ODB HECIG TO EQP table, type SCAN in the COMMAND field and perform a FIND by pressing the PF1 function key. If any valid HECIG codes are missing, an ADD must be performed.

NOTE 1 If a DS1 Access Point Type is missing from this table when a Carrier Record is issued from the TIRKS system to NSDB, the only way to get correct data to ITS is to reissue the Circuit Record from the TIRKS system to NSDB after the table entry is made. Available functions for the ODB Value Format Table are ADD, DELETE, and UPDATE (see BR 756–551–700, the *TTS Position Guide*, for detailed information about how to use these functions).

NOTE 2 In order for the multi-point circuits to be built correctly, this table must be populated with the HECIGs for all DDS bridges, and the ODB CO OPTIONS table fields PORTNAM1, PORTNAM2, and PORTNAM3 must contain the port parsing masks.

NOTE 3 To allow NSDB to recognize repeaters used on 56KB DDS circuits (these repeaters are entered on the 'P' and 'E*' lines of the circuit details), the HECIGs for all repeater plug-ins entered on the circuit details must be entered via this table.

NOTE 4 To enable NSDB to recognize OCU equipment, table entries must be established for all applicable equipment. NSDB derives the TABLE NAME from the first seven (7) characters of the HECIG value identified on the 'P' line of the Circuit Details. The table entry must indicate EQPTTYP = 'PL', and HECITYPE = 'OCU' or 'OCUDP'.

INTERFACE ID = TIRKS-NSDB & WFA/C–OPS/INE

This table is prepopulated with OPS/INE equipment codes by Bellcore. To view data on this table, follow these steps:

1. From a cleared screen, type "/FOR VJTDSO" and press the ENTER key.
2. On the returned screen, type "ODB HECIG TO EQP" in the TABLE NAME field.
3. Type "INE" in the TABLE KEY field.
4. Perform a FIND by pressing the PF1 function key.

### 7.17.4 Maintaining the Table

INTERFACE ID = TIRKS-NSDB-ITS/Digital

The NSDB System Administrator should coordinate with CPC personnel to identify changes to HECIG Group Codes or Equipment Types that NSDB is likely to encounter with regard to the TIRKS-NSDB-ITS/Digital interface. When changes occur that impact the HECIG, EQPTTYPE, or HECITYPE fields, the ODB HECIG TO EQP table should be modified. If this is not done, ITS/Digital testing can be affected.

INTERFACE ID = TIRKS-NSDB & WFA/C–OPS/INE

This table is prepopulated with data from Bellcore for OPS/INE–involved equipment. However, NSDB and OPS/INE system administrators should coordinate with CPC personnel to identify any new HECIGs that are used to equip the network elements that OPS/INE provisions. These will need to be added to the ODB HECIG TO EQP table. NSDB will not be able to identify the associated circuit data to pass to OPS/INE for any HECIGs that are not defined on this table.

## 7.18 ODB IMAGE

### 7.18.1 Table Description

This table is used to scan the service order image at completion time for the purpose of retrieving selected remarks sequences. These remarks are then stored for use when the line record is displayed during maintenance and installation activities.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB IMAGE          TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: VALUES USED TO SCAN THE SRV ORDER IMAGE  REL/LEV:            MOD: Y
   FIELD    FIELD
   NAME     VALUE
   RMK_SECT ---RMKS   *
   PRMK_FID RMK  *
   RRMK_FID RRMK *
   ASG_SECT ---ASGM   *
   SNE_SECT ---S&E    *
   BIL_SECT ---BILL   *
   CTL_SECT ---CTL    *
   DIR_SECT ---DIR    *
   TFC_SECT ---TFC    *
   IMG_DB   Y*
   CD_TYPE       *
   CD_LINE#   *
   CD_START   *
   CD_LNGTH   *

TTS210I FIND COMPLETED
```

**Figure 7–27.** ODB IMAGE Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBADO, VJDBCMO, VJDBCPO, VJDBIPM, VJDBL4P, VJDBPRS, VJDBUPO |

### 7.18.2 Field Descriptions

RMK_SECT

Full Name: Remarks Section

Definition: This field identifies to NSDB how the remarks section header appears on a service order.

Data Content: 10 A/N

Valid Entries: BCC provided; can vary by company.

Example: —RMKS

PRMK_FID

Full Name: Plant Remarks Field Identifier

Definition: This field identifies to NSDB how the plant remarks identifier appears on a service order. Plant remarks are usually associated with local facilities or equipment.

Data Content: 5 A/N

Valid Entries: BCC provided; can vary by company.

Example: PRMK

RRMK_FID

Full Name:

Definition: This field identifies to NSDB how the retained remarks identifier appears on a service order. Retained remarks can be anything deemed useful for future installation or maintenance activities.

Data Content: 5 A/N

Valid Entries: BCC provided; can vary by company.

Example: RRMK

ASG_SECT

Full Name: Assignment Section

Definition: This field identifies to NSDB how the Assignment Section identifier appears on a service order.

Data Content: 10 A/N

Valid Entries: BCC provided; can vary by company.

Example: —ASGM

SNE_SECT

Full Name: Service & Equipment Section

Definition: This field identifies to NSDB how the Service and Equipment Section identifier appears on a service order.

Data Content: 10 A/N

Valid Entries: BCC provided; can vary by company.

Example: —S&E

BIL_SECT

Full Name: Billing Section

Definition: This field identifies to NSDB how the Billing Section identifier appears on a service order.

Data Content: 10 A/N

Valid Entries: BCC provided; can vary by company.

Example: —BILL

CTL_SECT

Full Name: Control Section

Definition: This field identifies to NSDB how the Control Section identifier appears on a service order.

Data Content: 10 A/N

Valid Entries: BCC provided; can vary by company.

Example: —CTLL

DIR_SECT

Full Name: Directory Section

Definition: This field identifies to NSDB how the Directory Section identifier appears on a service order.

Data Content: 10 A/N

Valid Entries: BCC provided; can vary by company.

Example: —DIR

TFC_SECT

Full Name: Traffic Section

Definition: This field identifies to NSDB how the Traffic Section identifier appears on a service order.

Data Content: 10 A/N

Valid Entries: BCC provided; can vary by company.

Example: —TFC

IMG_DB

Full Name: Image Database

Definition: This field indicates whether or not the NSDB Image Database is to be loaded with Service Order images.

Data Content: 1 A

Valid Entries: Y (Yes, the Image database is to be loaded with service order images. Y is the default value.)

N (No, do not load the Image database.)

CD_TYPE

Full Name: Completion Date Type

Definition: This field defines the completion date as a fixed positional field or a floating field. If the date is not found, the system will use the current date as the completion date.

Data Content: 5 A

Valid Entries: FLOAT (use the date appearing after the '\CD' delimiter in the identification section of the service order image.)

FIXED (use entries in the CD_LINE#, CD_START, and CD_LNGTH fields to locate the completion date.)

CD_LINE#

Full Name: Completion Date Line Number

Definition: The line number in the fixed portion of the identification section on the service order image where the completion date can be found.

Data Content: 2 N

Examples: 8, 65

CD_START

Full Name: Completion Date Start

Definition: The starting position of the completion date on the specified line in the identification section of the service order image.

Data Content: to 2 N

Examples: 3, 16

CD_LNGTH

Full Name: Completion Date Length

Definition: The length of the completion date field.

Data Content: to 2 N

Examples: 8, 10

### 7.18.3 Creating the Table

Follow the steps below to create the ODB IMAGE table.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB IMAGE in the TABLE NAME field and press PF1 to FIND.
3. On the returned screen, in the RMK_SECT field, type in the header for the remarks section that is used on service orders that NSDB will encounter. As an example, if the remarks section header used by your company is '—— RMKS', then type —— RMKS in the RMK_SECT field.
4. In the PRMK_FID field, type in the header for the plant remarks that is used on service orders that NSDB will encounter. As an example, if the plant remarks header used by your company is 'PRMKS', then type PRMKS in the PRMK_FID field.
5. In the RRMK_FID field, type in the header for the retained remarks that is used on service orders that NSDB will encounter. As an example, if the retained remarks header used by your company is 'RRMKS', then type RRMKS in the RRMK_FID field.
6. In the ASG_SECT, SNE_SECT, BIL_SECT, CTL_SECT, DIR_SECT, and TFC_SECT fields, type in the headers that are associated with the assignment, service and equipment, billing, control, directory, and traffic sections.
7. In the IMG_DB field, enter a Y if your company wishes to store service order images in NSDB. If these images are not going to be stored, enter a N value.
8. When these fields are populated, ADD them by pressing PF4.

### 7.18.4 Maintaining the Table

Modifications to this table are required if the Section Identifiers on service orders change. The service order images can be saved, or not saved in the Image database depending on the entry in the IMG_DB field.

## 7.19 ODB LOCATIONS

### 7.19.1 Table Description

This table is used to determine whether OPS/INE should receive data associated with a particular network element. This check is performed for special service, message, and carrier orders.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB LOCATIONS     TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: NSDB LOCATIONS                             REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   LOCATION              *
   INE_SUPP  *
```

**Figure 7–28.** ODB LOCATIONS Table

| | |
|---|---|
| **Table Key(s):** | INE |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBICK |

## 7.19.2 Field Descriptions

LOCATION

Full Name: Location

Definition: The code representing a Network Element location, building location, city/state location, or a state location.

Data Content: to 11 A/N

Valid Entries: the 11–character Network Element location,
the 8–character building location,
the 6–character city/state location,
or the 2–character state location (the 5th and 6th characters of the Network Element location code)

INE_SUPPORTED

Full Name: Location Supported Indicator

Definition: This field specifies whether the location entry in the LOCATION field is supported by OPS/INE.

Data Content: 1 A

Valid Entries: Y (Yes, this is a supported location)

N (No, this is not a supported location)

### 7.19.3 Creating the Table

Follow the steps below to enter data in the ODB LOCATIONS table:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB LOCATIONS" in the TABLE NAME field, "INE" in the TABLE KEY field, and press the PF1 function key.
3. In the LOCATION field, type the appropriate location code to be defined for OPS/INE.
4. In the INE_SUPPORTED field, type a "Y" or "N" to specify whether the LOCATION entry is supported by OPS/INE.
5. Add this information to the ODB LOCATIONS table by pressing the PF4 function key.

NSDB will first check by the 11–character Network Element location code, and will proceed up the location hierarchy. Lower level entries may, therefore, be populated as either exceptions or rules.

> **NOTE –** The 8–character location entries are equivalent to the values on the TIRKS CMOPT screen, and may be copied over at the initial turnup of the OPS/INE interface.

### 7.19.4 Maintaining the Table

This table should be updated as revisions occur to the Network Element locations supported by OPS/INE.

## 7.20 ODB MCO OPTIONS

### 7.20.1 Table Description

This table is used to associate the Maintenance Control Office (MCO) with its error message LTERM device. Each MCO – LTERM combination is identified by a separate ODB MCO OPTIONS table.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB MCO OPTIONS   TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: ODB OPTIONS FOR THE MCO OR ADMIN AREA    REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   MCO_ADMN             *
   MSGLTERM           *
```

**Figure 7–29.** ODB MCO OPTIONS Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | MCO Values or Administrative Area Values |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBCPP, VJDBL1P, VJDMI1P |

### 7.20.2 Field Descriptions

MCO_ADMN

| | |
|---|---|
| Full Name: | Maintenance Control Office or Administrative Area |
| Definition: | The designated Maintenance Control Office (MCO) or Administrative Area that receives error messages. |
| Data Content: | Up to 11 A/N. |
| Valid Entries: | Any valid MCO or valid 2-digit Administrative Area code. |

MSGLTERM

| | |
|---|---|
| Full Name: | Message LTERM |
| Definition: | Logical Terminal code that identifies the destination for error messages associated with the MCO_ADMN entry above. |
| Data Content: | 8 A/N |
| Valid Entries: | Any valid LTERM code assigned by the BCC. |

### 7.20.3 Creating the Table

Follow the steps below to complete the NSDB MCO OPTIONS table.

1. Type /FOR VJTDSO and hit enter.
2. On the returned screen, type in ODB MCO OPTIONS in the TABLE NAME field and press PF1.
3. Type in the desired values for the MCO_ADMN and MSGLTERM fields.
4. Now perform an ADD by pressing PF4.

### 7.20.4 Maintaining the Table

When an error occurs in NSDB, three NSDB TTS tables are checked to determine the disposition of the error message. These tables are the *ODB EXCEPTIONS, ODB MCO OPTIONS,* and the *ODB CO OPTIONS* tables.

[1] NSDB looks into the ODB EXCEPTIONS table to see if this error message has any special processing set. This is the table that allows you to reroute or suppress a message. You can reroute or suppress a message for the entire NSDB system, or by individual MCO. (For the TIRKS data that is sent via the EDIIS ISSUWD event, the error message routing is performed using Administrative Area, all other interfaces use MCO.)

[2] If no instructions are found in the ODB EXCEPTIONS table, NSDB looks at the ODB MCO OPTIONS table to see if error routing has been turned on for all messages belonging to an MCO. Here you can specify routing for all messages for an MCO. (For the ISSUWD interface, routing is performed using Administrative Area, all other interfaces use MCO.)

[3] If no routing information is found in the above two tables, NSDB looks in the ODB CO OPTIONS table for the default error LTERM. This is the location that will receive the error message if no other routing was selected on the first two tables.

Any message that has been suppressed in the ODB EXCEPTIONS table will remain suppressed until the table is updated to stop the suppression. Periodic review of all suppressed messages is recommended so that important notifications are not lost.

## 7.21 ODB MON SWITCH

### 7.21.1 Table Description

The ODB Monitored Switch types table (ODB MON SWITCH) is a company level table used to filter trunk data received from the TIRKS system. This filtering was not needed previously because the only user of this data was the NMA system. Enhancements to the WFA/C system have resulted in that system's need for trunk data as well. WFA/C requires trunk data for switch types that the NMA system is not currently monitoring (#5 crossbar for example). In order to separate the trunk data that the NMA system needs (a subset of the WFA/C data), the ODB MON SWITCH table was developed.

The ODB MON SWITCH table uses switch universe codes to identify the switch types (and trunk data) that the NMA system needs. The universe codes come from the Generic Trunk Administration System (GTAS) used in the TIRKS system. Examples include 1ES for #1ESS switches, and DMS for the DMS 10 switches. By identifying these switch universe codes, NSDB can send the NMA system only the data it needs.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB MON SWITCH    TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                              # OF RECORDS: 0007
NOTE: NSDB SUBSYSTEM MONITORED SWITCH TYPES    REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   UNIV_CD     *
```

**Figure 7–30.** ODB MON SWITCH Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | GTAS Universe Codes for Switch Types |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.1 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBCHK |

### 7.21.2 Field Descriptions

UNIV_CD

Full Name: GTAS Universe Code

Definition: A 1 to 3 character code that identifies the type of switch associated with a trunk (as defined by the Generic Trunk Administration System (GTAS)).

Data Content: 1 to 3 A/N

Valid Entries: Other entries can be obtained by consulting the full list of GTAS Universal Codes, but the following codes are prepopulated by Bellcore:

1. 1ES
   Identifies the 1ESS™ and 1AESS switches.
2. 2A
   Identifies the AT&T 2A STP switch.
3. 2ES
   Identifies the 2ESS and 2AESS switches.
4. 5EC
   Identifies the 5ESS® combined switches.
5. DMC
   Identifies the DMS™ –100/200 combined switches.
6. DMS
   Identifies the DMS-10 switch.

---

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

### 7.21.3 Creating the Table

The ODB MON SWITCH TTS table is shipped from NSDB prepopulated with the GTAS universe codes for all switch types that the NMA system is currently capable of monitoring. The pre-populated UNIV_CD values are:

- 1ES
- 2A
- 2ES
- 5EC
- DMC
- DMS

### 7.21.4 Maintaining the Table

The prepopulated entries for universe code can be added to or deleted to meet any company–specific needs. If new codes are added to the ODB MON SWITCH table, they must also be assigned in the GTAS system. If the universe code does not appear in GTAS and in the ODB MON SWITCH table, the data *will not* be sent to the NMA system.

## 7.22 ODB OFFICE DATA

### 7.22.1 Table Description

This table contains the Point Of Termination (POT) locations for the DDS circuit base. This is,essentially, the last central office before the customer loop begins. The ODB OFFICE DATA table is used to validate the POT locations used for DDS circuits.

Each POT office CLLI that is subject to validation needs to be entered on this table in the OFFICE field. Coordination between the TIRKS, NSDB, and ITS system administrators is recommended in determining and maintaining the list of POT offices.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                  /FOR

TABLE NAME:  ODB OFFICE DATA   TABLE KEY:                  ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0004
NOTE: IDENTIFIES HUB AND POP OFFICES             REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   OFFICE               *
   TYPE       *
   HUB ID      *
```

**Figure 7–31.** ODB OFFICE DATA Table

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB OFFICE DATA   TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:  DLLSTXBZWAC                 # OF RECORDS: 0004
NOTE: IDENTIFIES HUB AND POP OFFICES          REL/LEV:           MOD: Y
   FIELD    FIELD
   NAME     VALUE
   OFFICE   DLLSTXBZWAC*
   TYPE     P*
   HUB ID     *




TTS210I FIND COMPLETED
```

**Figure 7–32.** Populated ODB OFFICE DATA Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Central Office CLLI Code Values |
| **Table Type:** | Validate |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | This table must be populated before running VJODU01. |
| **Module(s):** | VJODU01 |

### 7.22.2 Field Descriptions

OFFICE

Full Name: Office Code

Definition: The CLLI code of the POT central office.

Data Content: Up to 11 A/N

Example: DLLSTXTL

TYPE

Full Name: Office Type

Definition: Indicates the office type.

Data Content: 1 A/N

Valid Entries: P = Point Of Termination (POT) Office
H = Hub Office (for future use)

### 7.22.3 Creating the Table

Follow the steps below to complete the ODB OFFICE DATA table.

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type in ODB OFFICE DATA in the TABLE NAME field and press PF1.
3. Type in the CLLI value for the first POT central office in the OFFICE field, and a "P" in the TYPE field.
4. Now perform an ADD by pressing PF4.
5. You should receive an "ADD COMPLETED" message.
6. Now overtype the CLLI value displayed with the next POT central office in the OFFICE field.
7. Again perform an ADD by pressing PF4.
8. Continue these steps until all POT offices are added.

### 7.22.4 Maintaining the Table

Coordination between the TIRKS, NSDB, and ITS system administrators is recommended in determining and maintaining the list of POT offices.

## 7.23 ODB ROUTEBY

### 7.23.1 Table Description

INTERFACE ID = TIRKS-NSDB-NMA/FACILITIES
= TIRKS-NSDB-NMA/Pass–Through
= TIRKS-NSDB-NMA/SWITCH

NSDB uses this table in conjunction with the ODB DATA ROUTING table to distribute selected TIRKS data to specific NMA systems. The ODB ROUTEBY table maps the NMA system ID (NMA Facilities and NMA Pass–Through, NMA Switch, or NMA ISDN) to the desired routing method(s). The routing methods are identified in the ROUTEBY fields. Routing is done by either city and state in combination, or just by state. The city and state information comes from the CLLI code of the A and Z central office identity on the WORD document received from the TIRKS system. The completed ODB ROUTEBY table directs the TIRKS records bound for NMA Facilities and NMA Pass–Through, NMA Switch, or NMA ISDN, based on the first ROUTEBY criteria satisfied during processing. The appropriate services handled by NMA Facilities, NMA Pass–Through, and NMA Switch are based on internal criteria in NSDB. No action is required by the system administrator to define the services associated with NMA Facilities, NMA Pass–Through, or NMA Switch for this table. Only the routing method is determined by this table.

**INTERFACE ID = TIRKS–NSDB & WFA/C–OPSINE**

NSDB uses this table in conjunction with the ODB DATA ROUTING table to distribute selected TIRKS data to the OPS/INE system. The ODB ROUTEBY maps the OPS/INE system ID to the desired routing method. The routing methods are identified in the ROUTEBY fields. Routing is done by either city and state in combination, or just by state. The city and state information comes from the CLLI code of the A and Z central office identity on the WORD document received from the TIRKS system. The completed ODB ROUTEBY table directs the TIRKS records bound for OPS/INE based on the first ROUTEBY criteria satisfied during processing.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB ROUTEBY       TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0002
NOTE: NSDB SUBSYSTEM ROUTEBY TABLE               REL/LEV:          MOD: Y
   FIELD    FIELD
   NAME     VALUE
   SYSID              *
   ROUTEBY1             *
   ROUTEBY2             *
   ROUTEBY3             *
   ROUTEBY4             *
   ROUTEBY5             *
```

Figure 7–33. ODB ROUTEBY Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | INE, NMAFAC, NMAISDN, NMASWI |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBI01, VJDMI03 |

### 7.23.2 Field Descriptions

SYSID

Full Name: System Identification

Definition: Identifies the interface that the routing is associated with: OPS/INE, NMA Facilities and NMA Pass–Through, NMA Switch, or NMA ISDN.

Data Content: 12 A/N

Valid Entries: INE (This value identifies OPS/INE)

NMAFAC (This value identifies NMA Facilities and NMA Pass–Through)

NMASWI (This value identifies NMA Switch)

NMAISDN (This value identifies NMA ISDN)

ROUTEBY1 through ROUTEBY5

Full Name: Route By

Definition: This field defines the routing method to be used for the OPS/INE interface. If routing by the method in ROUTEBY1 fails, then ROUTEBY2 is used, etc. This will not result in duplicate routing. Two methods are available for routing:

- AGGCITYST is entered in the ROUTEBY1 field to indicate that the city/state combination derived from the CLLI of location A and the city/state combination derived from location Z is used.
- AGGST is entered in the ROUTEBY2 field to indicate that the state derived from the CLLI of location A and the state derived from the CLLI of location Z is used.

*OR*

This field defines the routing method to be used for NMA–related interfaces. If routing by the method in ROUTEBY1 fails, then ROUTEBY2 is used, etc. This will not result in duplicate routing. Two methods are available for routing:

- AZCITYST entered in the ROUTEBY field indicates that the city/state combination derived from the CLLI of location A and the city/state combination derived from location Z is used.
- AZST entered in the ROUTEBY field indicates that the state derived from the CLLI of location A and the state derived from the CLLI of location Z is used.

Data Content: 8 A

Valid Entries: For NMA–related interfaces, use AZCITYST and AZST

For the OPS/INE interface, use AGGCITYST and AGGST

### 7.23.3 Creating the Table

**INTERFACE ID = TIRKS-NSDB-NMA/FACILITIES**
**= TIRKS-NSDB-NMA/Pass–Through**
**= TIRKS-NSDB-NMA/SWITCH**

Three ODB ROUTEBY tables must be created if the NMA Facilities, NMA Pass–Through, and NMA Switch (including ISDN) interfaces are all used. The NMA Facilities and NMA Pass–Through interfaces use a single ODB ROUTING table with a SYSID of NMAFAC. The NMA Switch interface uses the ODB ROUTEBY table with the SYSID of NMASWI. If NMA ISDN is also used, then the SYSID is NMAISDN on the third ODB ROUTEBY table. The routing method used for NMA Switch is controlled separately from the method used by NMA Facilities and NMA Pass–Through and NMA ISDN.

To create the ODB ROUTEBY table, follow the steps below:

1. From a cleared screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB ROUTEBY in the TABLE NAME field. Leave the TABLE KEY field blank. Press the PF1 function key to FIND.
3. On the returned screen, type the desired interface identifier in the SYSID field. (Use NMAFAC for NMA Facilities and NMA Pass–Through, NMASWI for NMA Switch, and NMAISDN if ISDN is monitored by NMA.)
4. Type the routing method, or methods, desired in the ROUTEBY field(s). Remember that these fields are in hierarchical order. ROUTEBY1 will be used first, and the following ROUTEBY fields will be used in order if ROUTEBY1 fails.
5. After typing information in the SYSID and ROUTEBY fields, press the PF4 function key to ADD the values.

INTERFACE ID = TIRKS–NSDB & WFA/C–OPSINE

An ODB ROUTEBY table must be created to use the OPS/INE interface. The system ID is "INE" and must contain a table record key of "INE".

To create the ODB ROUTEBY table, follow the steps below:

1. From a cleared screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB ROUTEBY in the TABLE NAME field, and INE in the TABLE RECORD KEY field. Press the PF1 function key to FIND.
3. On the returned screen, type INE in the SYSID field.
4. Type AGGCITYST in the ROUTEBY1 field, and AGGST in the ROUTEBY2 field.
5. After typing information in the SYSID and ROUTEBY fields, press the PF4 function key to ADD the values.

### 7.23.4 Maintaining the Table

If the routing scheme requires adjustments, the ROUTEBY field(s) can be changed to accommodate most situations.

## 7.24 ODB SCVAL

### 7.24.1 Table Description

This table is used to refine the selections made in the ODB CTVAL table so that the online validation process will use only those circuits that are appropriate. Three fields on the ODB SCVAL table are used to refine the circuit types that are used. These fields are DDS, DS1, and CLFI. The tables below show the results of the possible entries made in ODB CTVAL and ODB SCVAL.

Special Services Circuits Online Validation Entries

**Table 7–3.** ODB SCVAL Online Validation Entries (1 of 2)

| IF THESE VALUES ARE ENTERED, | | THEN THE RESULT IS |
|---|---|---|
| ODB CTVAL | ODB SCVAL | |
| SPEC_SVC = Y | DDS = BLANK<br>= N<br>= Y<br>DS1 = BLANK<br>= N<br>= Y | ALL special services circuits are submitted for online validation*. |
| SPEC_SVC = N | DDS = Y<br>DS1 = BLANK<br>= NO | Only special services circuits that are DDS will be submitted for online validation. |
| SPEC_SVC = N | DS1 = Y<br>DDS = BLANK<br>= NO | Only special services circuits that are DS1 will be submitted for online validation. |

* ALL special services in this case means *both* DS1 and DDS; analog special services are not included in the validation process.

## INTERFACE ID = TIRKS-NSDB-ITS/Digital

Carrier Circuit Online Validation Entries

**Table 7-3.** ODB SCVAL Online Validations Entries (2 of 2)

| IF THESE VALUES ARE ENTERED, | | THEN THE RESULT IS |
|---|---|---|
| ODB CTVAL | ODB SCVAL | |
| CARRIER = Y | CLFI = BLANK<br>= N<br>= Y | ALL carrier circuits are submitted for online validation. |
| CARRIER = N | CLFI = Y | Only carrier circuits identified by CLFI will be submitted for online validation. |

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB SCVAL          TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0001
NOTE: ODB SVC CODE VAL. PERMISSIONS               REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   DDS       Y*
   DS1       Y*
   CLFI      Y*
   SPARE1     *
   SPARE2     *
   SPARE3     *
   SPARE4     *
   SPARE5     *
   SPARE6     *
   SPARE7     *



TTS210I FIND COMPLETED
```

**Figure 7–34.** ODB SCVAL Table

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB SCVAL          TABLE KEY: MIAMFLC1           ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0001
NOTE: ODB SVC CODE VAL. PERMISSIONS              REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   DDS       Y*
   DS1       Y*
   CLFI      Y*
   SPARE1     *
   SPARE2     *
   SPARE3     *
   SPARE4     *
   SPARE5     *
   SPARE6     *
   SPARE7     *



TTS210I FIND COMPLETED
```

**Figure 7–35.** Populated ODB SCVAL Table

| | |
|---|---|
| **Table Key(s):** | ALLMCO, MCO |
| **Table Record Key(s):** | MCO CLLI Values |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBB1P, VJDBLVP |

### 7.24.2 Field Descriptions

DDS

Full Name: Digital Data Service

Definition: Any service classified as a Digital Data Service by its service code in the CLCI standards. This field is used to include or exclude DDS for the online validation process.

Data Content: 1 A

Valid Entries: Y (Include DDS in online validation.)

N (Do not include DDS in online validation.)

blank (Same as N.)

DS1

Full Name: Digital Signal level 1 (1.544 MB/s)

Definition: Any service classified as a DS1 by its service code in the CLCI standards. This field is used to include or exclude DS1 for the online validation process.

Data Content: 1 A

Valid Entries: Y (Include DS1 in online validation.)

N (Do not include DS1 in online validation.)

blank (Same as N.)

CLFI

Full Name: Common Language Facility Identifier

Definition: The CLFI field is used to identify carrier circuits.

Data Content: 1 A

Valid Entries: Y (Include CLFI circuits in online validation.)

N (Do not include CLFI circuits in online validation.)

blank (Same as N.)

### 7.24.3 Creating the Table

The SCVAL table must be created with the CTVAL table in mind. As mentioned earlier, these two tables work together. The SCVAL table is used as a refinement to the broad categories selected in the CTVAL table.

Follow the steps below to create the SCVAL table.

1. From a clear screen, type /FOR VJTDSO and hit return.
2. On the returned screen, type in SCVAL in the TABLE NAME field, and press PF1 to FIND.
3. Make the desired Y or N entries in the DDS, DS1, and CLFI fields. (See table 7–3 to determine the results of these entries when combined with the CTVAL table entries.)
4. When the above fields are populated, press PF4 to ADD the values.

### 7.24.4 Maintaining the Table

If the set of circuits that require validation changes, both the ODB SCVAL and ODB CTVAL tables may need adjustments. See the table dependencies chart for possible variations.

## 7.25 ODB SDT DEFAULTS

### 7.25.1 Table Description

NSDB performs default processing whenever an ISDN service order is received from SOAC. Once it has been determined that default processing will take place, the system consults the ODM EXK TO SW ID table to determine the ISDN Switch Type. If the ISDN_DEF option is turned on in the ODB CO OPTIONS TTS table, then NSDB will generate the appropriate defaults based on user–defined entries in this table, ODB SDT DEFAULTS, and the ODB DEFAULT TAGS table. Defaults are stored in the Line Record database.

The ODB DEFAULT TAGS table is used to provide a list of candidate tags in support of ISDN Packet services translations when specific parameter values are not present on the ISDN service order. (This feature will extend to ISDN Voice services translations in the future.) Some of these tags will be present in the SOAC message, and some will not.

This table, ODB SDT DEFAULTS, contains default values for either the entire tag or for parameters within an existing tag value. If the tags are not present at all, then default values for the tags are obtained from this table. If the tag is present, then additional default parameter settings may need to be added to the tag value sent by SOAC and will be provided from this table.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB SDT DEFAULTS  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0000
NOTE: INDICATES PRIORITY SELECTION FOR FLC  REL/LEV:            MOD:
   FIELD     FIELD
   NAME      VALUE
   SOAC_TAG          *
   NO_TAG_D                                                       *
   PARCE_1               *
   NSDB_D1                                                        *
   PARCE_2               *
   NSDB_D2                                                        *
   PARCE_3               *
   NSDB_D3                                                        *
   PARCE_4               *
   NSDB_D4                                                        *
   PARCE_5               *
   NSDB_D5                                                        *
   PARCE_6               *
   NSDB_D6                                                        *
   PARCE_7               *
```

**Figure 7–36.** ODB SDT DEFAULTS Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB SDT DEFAULTS  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0000
NOTE: INDICATES PRIORITY SELECTION FOR FLC     REL/LEV:           MOD:
   FIELD     FIELD
   NAME      VALUE
   NSDB_D7                                                          *
   PARCE_8              *
   NSDB_D8                                                          *
   PARCE_9              *
   NSDB_D9                                                          *
   PARCE_10             *
   NSDB_D10                                                         *
   PARCE_11             *
   NSDB_D11                                                         *
   PARCE_12             *
   NSDB_D12                                                         *
   PARCE_13             *
   NSDB_D13                                                         *
   PARCE_14             *
   NSDB_D14                                                         *
```

**Figure 7–37.** ODB SDT DEFAULTS Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | Exchange Key or Switch Type |
| **Table Record Key(s):** | Tag name from ODT DEFAULT TAGS table |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | If the ODB SDT DEFAULTS table changes, you must also remember to check the first table retrieved in the process, ODB DEFAULT TAGS. |
| **Related Table(s):** | ODB DEFAULT TAGS |
| **Module(s):** | VJDBSDT |

### 7.25.2 Field Descriptions

SOAC_TAG

Full Name: SOAC Tag Name

Definition: The SOAC tag name for which defaults or additional default parameters are defined in the ODB SDT DEFAULTS table.

Data Content: 8 A

Example: LKLP

NO_TAG_D

Full Name: No Tag Default

Definition: The value to be used as a default when the tag in the SOAC_TAG field is not present in the SOAC message.

Data Content: 60 A/N

Example: TWO WAY

PARCE_1 through PARCE_14

Full Name: Parce Value

Definition: The pattern/parameter name the system should search for if the tag defined in the SOAC_TAG field is present in the SOAC message. Up to 14 Parce Values can be specified in this table.

Data Content: 12 A/N

Example: N2.

NSDB_D1 through NSDB_D14

Full Name: NSDB Default Value

Definition: The value to be appended to the SOAC tag name (based on the Parce Value) when the tag defined in the SOAC_TAG field is present in the SOAC message. Up to 14 Default Values can be specified in this table.

Data Content: 60 A/N

Example: N2.10

### 7.25.3 Creating the Table

Follow the steps below to complete the table:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type "ODB SDT DEFAULTS" in the TABLE NAME field.
3. Press the FIND PF1 function key.
4. Enter a TABLE KEY that is either an Exchange Key or a Switch Type, as appropriate. (The system will use the Exchange Key first in default processing, and if it is not found, will use the Switch Type.)
5. In the TABLE RECORD KEY field, type a candidate tag name that has been defined on the ODB DEFAULT TAGS table.
6. For each SOAC tag value, type entries as needed in the NO_TAG_D field, the PARCE_1 through PARCE_14 fields, and the corresponding NSDB_D1 through NSDB_D14 fields.
7. When all entries are made, ADD the values by pressing the ADD PF4 function key.

### 7.25.4 Default Processing Associated with the ODB SDT DEFAULTS Table

*Default for an entire tag/value*

If the candidate tag *is not* present in the SOAC message, the a default value is set from the ODB SDT DEFAULTS table using the value in the NO_TAG_D (No Tag Default) field.

*Adding additional default parameters to an existing tag/value pair*

If the candidate tag *is* present in the SOAC message, then the value will be parsed by NSDB to determine whether parameter names (defined in the PARCE_1 through PARCE_14 fields) are present. If parameter names are found, NSDB will not add a default value, but will continue on with the next Parce field in the list.

If a parameter name is not found, then NSDB will append the value in the NSDB_D* field to the end of the tag value sent by SOAC. This process continues until all PARSE_* and NSDB_D* pairs are processed.

## 7.26 ODB SEC TO SRC

This table is described for each INTERFACE ID that it supports. This is necessary because the ODB SEC TO SRC table is used differently for each INTERFACE ID. To find the appropriate description, locate the INTERFACE ID heading for the combination of systems that you are interested in. The table description for that INTERFACE ID follows the heading.

### 7.26.1 Table Description

**INTERFACE ID = TIRKS-NSDB-ITS/Digital**
**= TIRKS-NSDB-NMA/Facilities**
**= TIRKS-NSDB-NMA/Switch**

This table is used to identify the circuit Source Code (SOURCE) associated with each TIRKS System Entity Code (SEC). The SEC is a code that identifies a computer-based operating system. The SOURCE is a code that performs essentially the same function. SOURCE is merely a shortened version of the 8–character SEC. Each SEC or SOURCE uniquely identifies an application (e.g., TIRKS, WFA/C, NSDB, etc.). Since the SEC and SOURCE serve the same purpose, but are used independently, there must be a way to show their relationship. The ODB SEC TO SRC table is used to identify the one-to-one relationship between a SEC code and its corresponding SOURCE code. This table is called upon to perform cross–referencing when only the SEC or the SOURCE is known, and the other is needed. If multiple TIRKS systems are used, a separate ODB SEC TO SRC table is required for each TIRKS system. The SOURCE is assigned by the BCC.

**INTERFACE ID = WFA/C-NSDB-ITS/ISDN**
**= SOAC-NSDB-WFA/C**

The ODB SEC TO SRC table is used to associate the NSDB SEC code to a SOURCE. The SOURCE is a BCC–assigned, single alpha character used to identify NSDB in certain situations. If multiple NSDB machines are used, a separate ODB SEC TO SRC table is required for each NSDB machine.

**INTERFACE ID = SOAC-NSDB-WFA/C**

In addition to the NSDB SEC to SRC information required above, the SOAC-NSDB-WFA/C interface requires another ODB SEC TO SRC table. This second usage maps the SEC of the Service Order Processor (SOP) to a SOURCE. The SOURCE is a BCC–assigned, single alpha character used to identify the SOP during the processing of order information.

### INTERFACE ID = WFA/C-NSDB (Measurement)

The ODB SEC TO SRC table is used to associate the WFA/C SEC code to a SOURCE code for the maintenance portion of the Measurement database. The SOURCE is a BCC–assigned, single alpha character used to identify WFA/C when a trouble report is sent from WFA/C to NSDB. The SOURCE entry is then added to the 8–character trouble report number to make it unique. This is especially important if multiple WFA/C systems are sending to one measurement system. If multiple WFA/C machines are used, a separate ODB SEC TO SRC table is required for each WFA/C machine to avoid a collision at the Alias database.

**NOTE –** *Do not* change this entry once data has been received from WFA/C since the SOURCE field is part of the key to the Measurement database. Doing so will result in duplicate trouble reports if a post–closeout update, resend, or batch load is done.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB SEC TO SRC     TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0039
NOTE: SEC TO CIRCUIT SOURCE TRANSLATION    REL/LEV:              MOD: Y
   FIELD     FIELD
   NAME      VALUE
   SEC               *
   SOURCE     *
```

**Figure 7–38.** ODB SEC TO SRC Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | System Entity Codes (SEC Values) |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | The data content should be the same as the similar table in the Special Services Center (SSC). |
| **Module(s):** | VJDBL1P |

### 7.26.2 Field Descriptions

SEC

Full Name: System Entity Code

Definition: A code, defined via the common language standard, that identifies an Operations or Provisioning system. This field is used to identify the system for which a corresponding SOURCE value is needed. NSDB stores the source, and builds a source for every interface. The TIRKS system SEC and the NSDB SEC values used on this table must be the same as the 'MYSEC' entries of the TCM USER CONTROL tables for the TIRKS system and NSDB respectively. The value in this field is BCC–supplied.

Data Content: 8 A/N

Example: ZRITXSEC

SOURCE

Full Name: Circuit Source Code

Definition: A code used to identify the source of a circuit record. The SOURCE field represents the system identified by the SEC field above.

Data Content: 1 A

Valid Entries: Any single letter assigned by the BCC to represent the system identified by the SEC used above.

### 7.26.3 Creating the Table

The table is described for each INTERFACE ID that it supports. This is necessary because the ODB SEC TO SRC table is used differently for each INTERFACE ID. To find the appropriate description, locate the INTERFACE ID heading for the combination of systems that you are interested in. The description for creating a table for a particular INTERFACE ID follows the appropriate heading.

**INTERFACE ID = TIRKS-NSDB-ITS/Digital**
**= TIRKS-NSDB-NMA/Facilities**
**= TIRKS-NSDB-NMA/Switch**

The NSDB System Administrator must create the ODB SEC TO SRC table for each TIRKS system machine that communicates with NSDB. The SEC field on this table should be the same value used in the 'MYSEC' field of the TCM USER CONTROL table for the TIRKS system. The SOURCE field is a single letter, supplied by the BCC, that also identifies the TIRKS system indicated in the SEC field.

Follow the steps below to complete the ODB SEC TO SRC table.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type in ODB SEC TO SRC in the TABLE NAME field and press the PF1 function key.
3. An unpopulated ODB SEC TO SRC table is returned. Type the appropriate values in the SEC and SOURCE fields.
4. After the SEC and SOURCE field values are entered, press the PF4 function key to ADD the values.

CREATION OF THE ODB SEC TO SRC TABLE

**INTERFACE ID = WFA/C-NSDB-ITS/ISDN**
**= SOAC-NSDB-WFA/C**

The NSDB System Administrator must create the ODB SEC TO SRC table for each NSDB machine deployed. The SEC field on this table should be the same value used in the 'MYSEC' field of the TCM USER CONTROL table for NSDB. The SOURCE field is a single letter, supplied by the BCC, that also identifies the NSDB machine indicated in the SEC field.

Follow the steps below to complete the ODB SEC TO SRC table.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type in ODB SEC TO SRC in the TABLE NAME field and press the PF1 function key.
3. An unpopulated ODB SEC TO SRC table is returned. Type the appropriate values in the SEC and SOURCE fields.
4. After the SEC and SOURCE fields are entered, press the PF4 function key to ADD the values.

CREATION OF THE ODB SEC TO SRC TABLE

**INTERFACE ID = SOAC-NSDB-WFA/C**

The NSDB System Administrator must create the ODB SEC TO SRC table for each SOP machine that NSDB will encounter. The SEC field is BCC–provided. The SOURCE field is a single letter, supplied by the BCC, that also identifies the SOP machine indicated in the SEC field.

Follow the steps below to complete the ODB SEC TO SRC table.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB SEC TO SRC in the TABLE NAME field and press the PF1 function key.
3. An unpopulated ODB SEC TO SRC table is returned. Type the appropriate values in the SEC and SOURCE fields.
4. After the SEC and SOURCE fields are entered, press the PF4 function key to ADD the values.

CREATION OF THE ODB SEC TO SRC TABLE

**INTERFACE ID = WFA/C-NSDB (Measurement)**

The NSDB System Administrator must create the ODB SEC TO SRC table for each WFA/C system that sends measurement data via NSDB. The SEC field is BCC–provided. The SOURCE field is a single letter, supplied by the BCC, that also identifies the WFA/C system indicated in the SEC field.

Follow the steps below to complete the ODB SEC TO SRC table.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB SEC TO SRC in the TABLE NAME field and press the PF1 function key.
3. An unpopulated ODB SEC TO SRC table is returned. Type the appropriate values in the SEC and SOURCE fields.
4. After the SEC and SOURCE fields are entered, press the PF4 function key to ADD the values.

> **NOTE –** *Do not* change this entry once data has been received from WFA/C since the SOURCE field is part of the key to the Measurement database. Doing so will result in duplicate trouble reports if a post–closeout update, resend, or batch load is done.

### 7.26.4 Maintaining the Table

This table should require maintenance only if the SEC codes change or new copies of the application are added.

## 7.27 ODB SERVICE CODE

### 7.27.1 Table Description

This table, when used for ITS/Digital services, associates service codes with their corresponding service rate and service type. It is an optional table which is used when data is not forthcoming from TIRKS. The ODB SERVICE CODE table provides a means of validation in the event that a circuit needs to be tested. The service code value used in this table is the 3rd and 4th characters of the CLCI for DDS circuits. The service rate is the 'speed' or Baud rate(s) for the particular service code. The service type identifies the kind of service being provided by the circuit, such as 'DD' for digital data. The association of these characteristics is needed by ITS in order to properly perform accurate testing.

The service code for each ITS testable candidate must be entered and linked to its associated rate and type. For service codes that have multiple rates, this field may be left blank. When this is done, the online format VJDBCD can be used to apply the proper rate.

The ODB SERVICE CODE table is also used to validate ISDN circuits for NMA/Switch, when ISDN circuits are monitored. Verification of designed ISDN circuits is based on the mapping of ISDN Service Codes (CLCI positions 3 & 4) to a Service Type value with a first character of "I". Each valid ISDN Service Code must be associated with a Service Type value beginning with "I" in the SVC TYPE field.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB SERVICE CODE  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0022
NOTE: SERVICE CODE MAPPING TO RATE AND TYPE      REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   SVCCODE     *
   SVCRATE       *
   SVCTYPE     *
```

**Figure 7–39.** ODB SERVICE CODE Table

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB SERVICE CODE  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:  DW                          # OF RECORDS: 0022
NOTE: SERVICE CODE MAPPING TO RATE AND TYPE    REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   SVCCODE   DW*
   SVCRATE   56   *
   SVCTYPE   DD*









TTS210I FIND COMPLETED
```

**Figure 7–40.** Populated ODB SERVICE CODE Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | DDS and ISDN Servie Code Values |
| **Table Type:** | Validate |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | This table must be populated before using NSDB. |
| **Module(s):** | VJDBMGR, VJODU01 |

### 7.27.2 Field Descriptions

SVCCODE

Full Name: Service Code

Definition: This field is used to identify the Service Codes for DDS circuits that are candidates for ITS testing. (A Service Code is the combination of the 3rd and 4th characters of the CLCI.)

The Service Code field is also used to identify valid service codes for ISDN circuits for NMA/Switch monitoring of ISDN circuits.

Data Content: 2 A/N

Valid Entries: Any valid Service Code provided by the BCC.

SVCRATE

Full Name: Service Rate

Definition: This field is used to associate the Service Rate with the Service Code entered above. (A Service Rate is the speed at which data can be passed, measured in kilobits or megabits per second.)

(The Service Rate field is left blank for the NMA/Switch usage.)

Data Content: 5 N

Valid Entries: 1.544, 2.4, 4.8, 9.6, 56, or blank.

**NOTE –** Blanks should be used only for Service Codes that represent multiple Service Rates. If the SVCRATE field is left blank, the RATE field on the Circuit Details administrative format, (VJDBCD), can be used to apply the proper rate. If the SVCRATE field is populated, then the VJDBCD format can not be used in this manner because the RATE field is protected.

SVCTYPE

Full Name: Service Type

Definition: This field is used to associate the service code entered in the SVCCODE field to a generic identity that is usable by the ITS system.

The Service Type field is also used to identify ISDN service types for NMA/Switch monitoring of ISDN circuits.

Data Content: 2 A/N

Valid Entries: DD (Identifies Service Codes that are for Digital Data.)

DS (Identifies Service Codes that are for Digital Signal level 1, i.e., 1.544 Mb/s.)

I* (Where I* is any valid ISDN Service Type. The "I" is used to indicate ISDN for circuit verification for NMA/Switch.)

### 7.27.3 Creating the Table

For **ITS/Digital usage,** follow the steps below to complete the ODB SERVICE CODE table.

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type in ODB SERVICE CO in the TABLE NAME field and press PF1.
3. On the returned screen, type in the service code in the SVCCODE field. You must also type in the service code's associated service rate (SVCRATE) and service type (SVCTYPE).
4. When the SVCCODE, SVCRATE, and SVCTYPE fields are typed in, you must perform an ADD by pressing PF4.
5. Repeat steps 3 and 4 until all candidate service codes are entered. You may check your completed work by typing SCAN in the COMMAND field and pressing the PF1 function key. All the service codes that exist on this table will be displayed on the returned screen.

For **NMA/Switch usage,** follow the steps below to complete the ODB SERVICE CODE table.

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type in ODB SERVICE CO in the TABLE NAME field and press PF1.
3. On the returned screen, type in the ISDN service code in the SVCCODE field. Leave the service rate (SVCRATE) field blank.
4. Type the ISDN Service Type in the SVCTYPE field.
5. Now perform an ADD by pressing PF4.
6. Repeat steps 3,4 and 5 until all candidate service codes and service types are entered. You may check your completed work by typing SCAN in the COMMAND field and pressing the PF1 function key. All the service codes that exist on this table will be displayed on the returned screen.

## 7.28 ODB USOC OPTIONS

### 7.28.1 Table Description

The ODB USOC OPTIONS table is an optional table that provides special processing to decrement the CKL value of a nondesigned circuit ID based on a user–identified Class of Service USOC. This processing is needed when the SOAC system provisions the nondesigned circuit with a CKL value of "1", and the user systems (WFA/C, LMOS) require the same circuit with a CKL/DPA value of "0".

In companies where the CKL difference exists, the ODB USOC OPTIONS table can be used to convert the CKL 1 value received from SOAC to a CKL 0 value used by WFA/C during maintenance activities. The special processing in NSDB is triggered when the USOC_OPT field on the ODB CO OPTIONS table is set to "Y". This causes the ODB USOC OPTIONS table to be called. The ODB USOC OPTIONS table contains the Class of Service USOCs that are subject to CKL decrementation. An additional field, CKL_1, is used to turn on / off the processing for its associated Class of Service USOC.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB USOC OPTIONS  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: ODB USOC OPTIONS                           REL/LEV:          MOD:
   FIELD     FIELD
   NAME      VALUE
   CSU         *
   CKL_1      *
```

**Figure 7–41.** ODB USOC OPTIONS Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Class of Service USOCs |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.1 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBMIS |

### 7.28.2 Field Descriptions

CDU

Full Name: Class of Service USOC

Definition: The Class of Service USOC that requires special processing to decrement the associated CKL value by 1.

Data Content: 3 A/N

CKL_1

Full Name: CKL Decrement Flag

Definition: This flag is used to turn on or off the special processing used for the associated Class of Service USOC entered in the CSU field.

Data Content: 1 A

Valid Entries: Y = Yes, decrement CKL value by 1.
N, or blank = No, do not decrement CKL value for this CSU.

### 7.28.3 Creating the Table

To create the ODB USOC OPTIONS table, follow the steps below:

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB USOC OPTIONS in the TABLE NAME field and press PF1 to FIND.
3. On the returned screen, type in the first Class of Service USOC value in the CSU field.
4. Type "Y" in the CKL_1 field to indicate CKL decrementation processing is to take place for the CSU entered above.
5. Press PF4 to ADD the values.
6. Repeat steps 3, 4, and 5 until all desired CSUs are designated for special CKL processing.

### 7.28.4 Maintaining the Table

This table is dependent upon the USOC_OPT field on the ODB CO OPTIONS table. The USOC_OPT field serves as the on/off switch for the ODB USOC OPTIONS table. Both tables should be considered when making changes.

## 7.29 ODB VALID FORMAT

### 7.29.1 Table Description

**USED FOR ADMINISTRATION OF NSDB ON-LINE FORMATS**

This table is used to identify the formats that can perform the JUMP/FIND function. The JUMP/FIND function uses the data currently displayed on a format to 'find' the desired view on a second format. The 'jump' is accomplished by typing the second format name in the /FOR field of the original displayed format, and pressing the ENTER key. The returned screen shows the second format (from the /FOR field) populated with data found by using the original format's data contents.

The ODB VALID FORMAT table can be used to display the destination or second formats as described above, for each NSDB online format. Information in this table can be added to, deleted, or updated as desired. The SCAN command can be used to list all the 'jump-to' formats at once.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB VALID FORMAT  TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                # OF RECORDS: 0255
NOTE: VALID NSDB FORMATS FOR ODB JUMP AND JUMP REL/LEV: 3.1.1.2   MOD: N
   FIELD    FIELD
   NAME     VALUE
   NSDBFMT          *
   SUBSYS           *
   TITLE                                                        *
```

**Figure 7–42.** ODB VALID FORMAT Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Valid Format Names |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore and BCC |
| **Updating Cautions:** | None |
| **Format(s):** | VJDBCD, VJDBMP |
| **Module(s):** | VJDBCDP, VJDBMPP |

### 7.29.2 Field Descriptions

NSDBFMT

Full Name: Format Name

Definition: Identifies the format name that accomodates the JUMP/FIND function.

Data Content: 8 A/N

Valid Entries: Valid Format Names

**NOTE –** Field is populated by Bellcore and the BCC.

SUBSYS

Full Name: Subsystem

Definition: The name of the subsystem to which the format belongs.

Data Content: 8 A/N

Valid Entries: Valid Subsystem

**NOTE –** Field is populated by Bellcore and the BCC.

TITLE

Full Name: Description or title of the format.

Definition: A description or title for the valid format.

Data Content: 50 A/N

Valid Entries: Valid description or title.

**NOTE –** Field is populated by Bellcore and the BCC.

### 7.29.3 Displaying the Table

The steps below describe how this table is most often used.

To determine the subsystem and description of an NSDB format:

1. Display the ODB VALID FORMAT table by typing /FOR VJTDSO and hit return. On the returned screen type in ODB VALID FORMAT in the TABLE NAME field and hit return.
2. Type in the format name you would like to know about in the TABLE RECORD KEY field, and press PF1.
3. The format name will appear in the NSDBFMT field, its subsystem name appears in the SUBSYS field, and a description of the format will appear in the TITLE field.

To view all the valid JUMP/FIND formats:

1. Display the ODB VALID FORMAT table as described above.
2. Enter SCAN in the COMMAND field.
3. Press PF1. All the current valid format names for the JUMP/FIND function are displayed.

## 7.30 ODB WCLLI – WCTR

### 7.30.1 Table Description

The ODB WCLLI – WCTR table is used to associate central office CLLI codes to their SOAC wire center codes. This mapping is used when the NSDB Line Record database is created from existing CIMAP/SSC data. The needed SOAC wire center code is not part of the data stored by CIMAP/SSC. The central office CLLI code however, is stored by CIMAP/SSC. During the Line Record creation process, the central office CLLI code is used to derive the SOAC wire center code by accessing the ODB WCLLI – WCTR table.

**NOTE –** You must type spaces on either side of the hyphen when accessing this table; i.e., "ODB WCLLI (space) – (space) WCTR".

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODB WCLLI - WCTR  TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0013
NOTE: TRANSLATES WIRE CENTER CLLI TO FAC WCTR  REL/LEV:            MOD: N
   FIELD     FIELD
   NAME      VALUE
   WCLLI               *
   WCTR           *
```

**Figure 7–43.** ODB WCLLI – WCTR Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Central Office CLLI Code Values |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | None |

### 7.30.2 Field Descriptions

WCLLI

| | |
|---|---|
| Full Name: | Wire Center Common Language Location Identification |
| Definition: | This field is used to identify the central office locations that correspond to SOAC wire centers. |
| Data Content: | 11 A/N |
| Valid Entries: | Any valid BCC–provided CLLI that maps to a SOAC wire center. |

WCTR

| | |
|---|---|
| Full Name: | SOAC Wire Center |
| Definition: | The SOAC wire center number that corresponds to the CLLI entry in the WCLLI field above. |
| Data Content: | 6 N |
| Valid Entries: | Any valid BCC–provided SOAC wire center number. |

### 7.30.3 Creating the Table

A ODB WCLLI – WCTR table must be created for each unique central office CLLI that NSDB is expected to encounter. The central office CLLI is mapped to the equivalent SOAC wire center number. A single SOAC wire center number may include several central office CLLI codes. Follow the steps below to create these tables.

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODB WCLLI – WCTR in the TABLE NAME field, and press PF1 to FIND.
3. On the returned screen, type the first central office CLLI code in the WCLLI field.
4. Type the SOAC wire center code in the WCTR field.
5. ADD the values by pressing PF4.
6. Type in the next CLLI code in the WCLLI field for the SOAC wire center displayed in the WCTR field, and press PF4.
7. Repeat step 6 until all the CLLI codes associated with the displayed SOAC wire center are completed.
8. Type the next SOAC wire center in the WCTR field, and enter the associated CLLI codes as in step 6.

### 7.30.4 Maintaining the Table

The SOAC, NSDB, and WFA/C system administrators should work together to maintain the integrity of this table.

# 7.31 ODM EXK TO SW ID

## 7.31.1 Table Description

This table is used in conjunction with the ODM TOS TO EXK table. The purpose of the ODM EXK TO SW ID table is to link or map an area code and prefix to a type for the associated central office switching machine. This is needed because the type of switching machine influences ITS auto-test processing. The combination of an area code and prefix comprise the Exchange Key (EXK). Central office location also relates to Exchange Key to switch mapping. This table identifies the CLLI code of the specified switching equipment.

The ODM EXK TO SW ID table is used internally by NSDB to map the Exchange Key to the switch type and CLLI code so that ITS and NMA can receive correct ISDN information for testing and monitoring purposes.

With NSDB Release 3.2, a field has been added to specify the central office type. This field is used to translate the switch type (which is based on the exchange key) to a one–character representation of the switch. This will be used by a variety of the measurement systems (MTAS, STAT, or TREAT) employed by the Bell Client Companies.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODM EXK TO SW ID  TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0035
NOTE: EXK KEY MAPPING TO SWITCH CLLI AND TYPE  REL/LEV:            MOD: N
   FIELD    FIELD
   NAME     VALUE
   EXK_KEY        *
   SW_CLLI         *
   SW_TYPE         *
   COTYPE    *




TTS210I FIND COMPLETED
```

**Figure 7–44.** ODM EXK TO SW ID Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Exchange Key Values |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDBMDS, VJDMO1P |

### 7.31.2 Field Descriptions

EXK_KEY

Full Name: Exchange Key

Definition: This field is used to identify the exchange key(s) that provide ISDN services which ITS can test.

Data Content: 6 A/N

Valid Entries: Any BCC–supplied exchange key that provides ISDN services which are ITS–testable, or that can be monitored by NMA/Switch.

SW_CLLI

Full Name: Common Language Location Identification for a Switch

Definition: A CLLI is the standardized code used to uniquely identify a location. This field is used to identify the location of a central office switch that provides ISDN service which ITS can test, or that can be monitored by NMA/Switch. A suffix of "CGx" indicates an electronic central office, where "x" designates a specific location number in the event of multiple electronic central offices. A suffix of "MGx" indicates a non–electronic central office, where "x" designates a specific location number in the event of multiple non–electronic central offices.

Data Content: 11 A/N

Valid Entries: Any CLLI that represents a central office switch providing ISDN services.

SW_TYPE

Full Name: Switch Type

Definition: This field identifies the type of central office switch that is associated with the CLLI and EXK_KEY field values described above.

Data Content: 10 A/N

Valid Entries: 5ESS
DMS100

COTYPE

Full Name: Central Office Type

Definition: This field is used to translate the switch type (which is based on the exchange key) to a one–character representation of the switch. This will be used by a variety of the measurement systems for maintenance data (MTAS, STAT, or TREAT) employed by the Bell Client Companies.

Data Content: 1 A/N

Valid Entries: The following table is used for the translation from switch type to Central Office Type. Users can define other switch types via control card options for the batch run VJCDM13, which is used in conjunction with the VJCSM04 and VJCSC08. For information about these runs, refer to chapter 8 in this Guide "Recommended Runs and Their Usage".

| COTYPE | SW_TYPE | DESCRIPTION |
|---|---|---|
| 1 | SXS | Step–by–Step |
| 2 | PAN | Panel |
| 3 | 1XB | No. 1 Crossbar |
| 4 | 5XB | No. 5 Crossbar |
| 5 | ESS | No. 1 ESS (2–wire) |
| 6 | 2ES | No. 2 ESS |
| 7 | LSG | Step–by–Step (Line Switch Group) |
| 8 | 1BC | No. 1 Crossbar with Common Choice |
| 9 | 4WE | No. 1 ESS (4–wire) |
| A | 3ES | No. 3 ESS |
| B | 3XB | No. 3 Crossbar |
| C | NSWEC | Nonstandard Western Electric Co. |
| D | RSS | Remote Switching System |
| E | 5ESS | No. 5 ESS |

### 7.31.3 Creating the Table

The NSDB system administrator should create this table with the companion table, ODM TOS TO EXK, in mind. To create the ODM EXK TO SW ID table, the system administrator needs to know the exchange keys providing ISDN services, the corresponding CLLIs of central office switching equipment, the types of switching equipment, and the types of central office equipment. The TABLE KEY for this table is blank and the TABLE RECORD KEY is the Exchange Key (EXK).

Follow the steps below to complete the ODM EXK TO SW ID table:

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type ODM EXK TO SW ID in the TABLE NAME field, and press the PF1 function key to FIND.
3. On the returned screen, type the appropriate values for the EXK_KEY, SW_CLLI, SW_TYPE, and COTYPE fields.
4. When all values are filled in, ADD the information by pressing the PF4 function key.

The ODM EXK TO SW ID can also be created/loaded mechanically. This method involves the extraction of data from the COSMOS Entity Summary table, the NSDB ODB WCLLI – WCTR TTS table, and the TIRKS System LOCREFDD database. A summary of the process follows:

1. COSMOS creates an output file containing the COSMOS Wire Center, Primary NPA/NXX, and EXK. The file is transferred to NSDB. NSDB extracts the exchange key (EXK). (EXK is a six–character numeric field that represents the NPA and NXX handled by the switch.)
2. NSDB extracts the NPA and eight–character CLLI code from the ODB WCLLI – WCTR TTS table. This data is used with TIRKS data to determine the Master Switch CLLI code.
3. The Master Location, the last three positions of the Sublocation(s), and the Switching System is extracted from the TIRKS LOCREFDD database.
4. The TIRKS data is compared to the extracted COSMOS and NSDB data to identify the needed fields for the ODM EXK TO SW ID table.
5. After the TIRKS, COSMOS, and NSDB data is validated and merged, a file is created. This file is used to load the ODM EXK TO SW ID table with EXK data from COSMOS used in the EXK_KEY field, Master Location data from TIRKS used in the SW_CLLI field, and Switching System data from TIRKS used in the SW_TYPE field.

Using this mechanized method requires that:

- the COSMOS tables are populated.
- COSMOS Release 17.2.6.3 or higher is implemented.
- the NSDB ODB WCLLI – WCTR table is populated.
- NSDB Release 3.1 or higher is implemented.
- the TIRKS System LOCREFDD database is populated.
- TIRKS System Release 16.0.1 or higher is implemented.
- NSDB will receive a unique EXK for each switch from COSMOS.
- the EXK normally represents a NPA/NXX found in the switch.
- the ODM EXK TO SW ID table is used with NMA Release 3.3 or higher.
- the ODM EXK TO SW ID table is used with ITS Release 3.0 or higher.

### 7.31.4 Maintaining the Table

Changes, additions, or deletions to the relationships between the Exchange Keys, Central Office CLLI codes, Switch Types, and Central Office Types should be reflected in the ODM EXK TO SW ID table. Coordination between the TIRKS, COSMOS, and NSDB administrators is recommended.

## 7.32 ODM TOS TO EXK

### 7.32.1 Table Description

This table is used by NSDB as part of the processing that occurs in the ITS Auto-test feature for ISDN circuits. This table identifies the relationship between the Testing Operations System (TOS) and its associated Exchange Keys (EXK). TOS is a generic term for systems that provide testing functions. In this case ITS is the TOS. The exchange key represents the area code and prefix of the central office switching machine where the ISDN circuit resides. This relationship between the TOS and the EXK is basic information required by the Operations Data Manager (ODM) subsystem of NSDB.

This table is used in conjunction with the ODM EXK TO SW ID table. The ODM EXK TO SW ID table associates the exchange key to the type of central office switch that supports the ISDN circuits that ITS will be testing.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODM TOS TO EXK    TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                              # OF RECORDS: 0003
NOTE: MAPPING OF EXCHANGE KEY TO TOS          REL/LEV:           MOD: N
   FIELD     FIELD
   NAME      VALUE
   TOS                *
   EXK01              *
   EXK02              *
   EXK03              *
   EXK04              *
   EXK05              *
   EXK06              *
   EXK07              *
   EXK08              *
   EXK09              *
   EXK10              *
   EXK11              *
   EXK12              *
   EXK13              *
   EXK14              *
```

**Figure 7–45.** ODM TOS TO EXK Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ODM TOS TO EXK    TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0003
NOTE: MAPPING OF EXCHANGE KEY TO TOS             REL/LEV:          MOD: N
   FIELD    FIELD
   NAME     VALUE
   EXK15          *
   EXK16          *
   EXK17          *
   EXK18          *
   EXK19          *
   EXK20          *
   EXK21          *
   EXK22          *
   EXK23          *
   EXK24          *
   EXK25          *
   EXK26          *
   EXK27          *
   EXK28          *
   EXK29          *
```

**Figure 7–46.** ODM TOS TO EXK Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | ISDN |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Module(s):** | VJDMMRM |

### 7.32.2 Field Descriptions

TOS

Full Name: Testing Operations System

Definition: A general term describing an operations system that provides testing functionality. This field currently is used to identify ITS/ISDN only.

Data Content: 8 A/N

Valid Entries: ISDN

EXK01 through EXK29

Full Name: Exchange Key

Definition: This field is used to identify the exchange key(s) that provide ISDN services that ITS can test.

Data Content: 6 A/N

Valid Entries: Any BCC–supplied exchange key that provides ISDN services which are ITS–testable.

### 7.32.3 Creating the Table

This table is used in conjunction with the ODM EXK TO SW ID table. The NSDB System Administrator must create both tables if ITS will be testing ISDN basic rate circuits. To create this table, the System Administrator must know the exchange keys that support the ISDN circuits that ITS will be testing.

Follow the steps below to complete the ODM TOS TO EXK table.

1. Type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type in ODM TOS TO EXK in the TABLE NAME field and press PF1.
3. Type in the appropriate values for the TOS and EXK fields.
4. When complete, ADD the values to the table by pressing PF4.

### 7.32.4 Maintaining the Table

New Exchange Keys that are candidates for ITS/ISDN testing should be added to this table.

## 7.33 OQS DB2 TABLES

### 7.33.1 Table Description

This TTS table, OQS DB2 TABLES, allows OQS queries against DB2 tables. It maps the OQS database name to the actual DB2 table name, and includes the high–order node of the DB2 table (OWNERID) to permit SQL statements to be built.

**NOTE** – The only field in this table that should ever be changed by the BCC is the OWNERID field. The other two fields, DATABASE and DB2TABLE, are populated by Bellcore.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  OQS DB2 TABLES    TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0010
NOTE: DB2 TABLES QUERIABLE BY OQS                REL/LEV: 4.1.1.3   MOD: N
   FIELD    FIELD
   NAME     VALUE
   DATABASE          *
   DB2TABLE                    *
   OWNERID           *
```

**Figure 7–47.** OQS DB2 TABLES Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | OQS DATABASE name that maps to a DB2 table |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.2 |
| **Data Supplied by:** | Bellcore and BCC |
| **Updating Cautions:** | The only field that should be changed by the BCC is the high–order node (OWNERID). The DATABASE and DB2TABLE fields should not change. |
| **Associated Subsystem:** | OQS |
| **Format(s):** | VJMIIQ, VJMIOQ, VJMMBQ, VJMMCQ, VJMMFQ, VJMMHQ, VJMMMQ, VJMMMNQ, VJMMSQ |
| **Run(s):** | VJOQR01, VJOQR04 |
| **Module(s):** | VJOQSII, VJOQDB2, VJOQSB2 |

### 7.33.2 Field Descriptions

DATABASE

| | |
|---|---|
| Full Name: | Database Name |
| Definition: | Name (shown on query screen) of the OQS database that maps to the DB2 table name. This field is shipped by Bellcore with the required information. |
| Data Content: | 8 A/N |
| Valid Entries: | Valid OQS database name |

DB2TABLE

| | |
|---|---|
| Full Name: | DB2 Table Name |
| Definition: | The actual name of the DB2 table that maps to the OQS database name. This field is shipped by Bellcore with the required information. |
| Data Content: | 18 A/N |
| Valid Entries: | Valid DB2 database name |

OWNERID

Full Name: Owner ID

Definition: The high–order node for a given DB2 environment.

Data Content: 8 A/N

### 7.33.3 Displaying the Table

1. Display the TTS table access screen by typing /FOR VJTDSO.
2. In the TABLE NAME field, type OQS DB2 TABLES.
3. Leave the TABLE KEY field blank, and enter a DATABASE value in the TABLE RECORD KEY field.
4. Press the PF1 function key to find the desired data.

### 7.33.4 Maintaining the Table

To update an entry, enter a C to the left of the field(s) being modified, make the appropriate modifications, and press the PF5 function key.

**NOTE** – The only field in this table that should ever be changed by the BCC is the OWNERID field.

## 7.34 OQS USERIDS

### 7.34.1 Table Description

**USED FOR CONTROLLING ACCESS TO OQS**

This table is used by the NSDB System Administrator to authorize permission for users of the Open Query System (OQS). The OQS USERIDS table is used in conjunction with the security procedures described in the *NSDB Security Administrator's Guide*, BR 190-534-003.

Currently, OQS can be used to initiate report queries from the TCM Message Administration (MA) databases supporting NSDB, as well as the NSDB databases that store both nondesigned and designed circuit information. These reports include both 'canned' queries, used for the TCM databases, and user–created queries for the NSDB nondesigned and designed databases, and the TCM Message Administration (MA) databases.

For detailed information about the use of OQS for NSDB, refer to the *Open Query System (OQS) User Manual* for NSDB, BR 190-534-300.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  OQS USERIDS       TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0010
NOTE: VALID USER IDS FOR OQS                     REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   USERID            *
```

**Figure 7–48.** OQS USERIDS Table

| | |
|---|---|
| **Table Key(s):** | Refer to 190–534–003 |
| **Table Record Key(s):** | Refer to 190–534–003 |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore and BCC |
| **Updating Cautions:** | None |
| **Associated Subsystem:** | OQS |
| **Format(s):** | VJS1ADM |
| **Module(s):** | VJS1U01 |

### 7.34.2 Field Descriptions

USERID

| | |
|---|---|
| Full Name: | User Identifier |
| Definition: | This field displays the BCC–assigned ID of the authorized OQS users for the NSDB product. |
| Data Content: | 8 A/N |
| Valid Entries: | Any valid BCC–authorized ID |

### 7.34.3 Creating the Table

See the *NSDB Security Administrator's Guide*, BR 190-534-003.

### 7.34.4 Maintaining the Table

See the *NSDB Security Administrator's Guide*, BR 190-534-003.

## 7.35 S1 EXITS

### 7.35.1 Table Description

The S1 EXITS table contains the S1 Exits routine transactions for the application program and is used to further interrogate security calls by the application programs. This table is not currently used, but is reserved for future use.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  S1 EXITS          TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: SECURITY EXIT TRANSACTIONS                  REL/LEV:          MOD: Y
   FIELD    FIELD
   NAME     VALUE
   SUBSYS           *
   TRANS            *
```

**Figure 7–49.** S1 EXITS Table

| | |
|---|---|
| **Table Key(s):** | not used |
| **Table Record Key(s):** | not used |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore (locked table) |
| **Associated Subsystem:** | S1 Security |
| **Updating Cautions:** | None |
| **Module(s):** | None |

### 7.35.2 Field Descriptions

SUBSYS

Full Name: Subsystem

Definition: Not currently used.

TRANS

Full Name: Transaction Code

Definition: Not currently used.

### 7.35.3 Creating the Table

This table is not currently used.

### 7.35.4 Maintaining the Table

This table is not currently used.

## 7.36 SOAC TO SOPID

### 7.36.1 Table Description

The SOAC TO SOPID TTS table associates the SOAC System Entity Code (SEC) to the SOPID in order to obtain the source of a given service order.

If this table is populated and the SOAC SEC is found, NSDB will use the SOPID to retrieve the service order source from the ODB SEC TO SRC TTS table. If the table is not populated or the TABLE RECORD KEY is missing, NSDB will use as the default the SOPID from the SOAC FCIF message (OCTL aggregate). No error conditions will be generated.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  SOAC TO SOPID     TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: ODB SOAC-SEC TO SOPID                      REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   SOACID            *
   SOPID             *
```

**Figure 7–50.** SOAC TO SOPID Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | SOAC SEC from the ROUTECTL TCM message |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.1.1.6 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | None |
| **Associated Subsystem:** | ODB |
| **Format(s):** | None |
| **Table(s):** | None |
| **Run(s):** | None |
| **Module(s):** | VJDBL5P |

### 7.36.2 Field Descriptions

SOACID

| | |
|---|---|
| Full Name: | SOAC Identifier |
| Definition: | Identifies the SEC (System Entity Code) of the SOAC involved in the origination of the service order. |
| Data Content: | 8 A/N |
| Example: | SOAC861 (no validations) |

SOPID

| | |
|---|---|
| Full Name: | Service Order Processor Identifier |
| Definition: | Identifies the SOP that transmitted the service to SOAC. |
| Data Content: | 8 A/N |
| Example: | SOP (no validations) |

### 7.36.3 Creating the Table

1. Display the TTS table access screen by typing /FOR VJTDSO.
2. In the TABLE NAME field, type "SOAC TO SOPID".
3. Leave the TABLE KEY and TABLE RECORD KEY fields blank.
4. Press the PF1 function key.
5. Enter the appropriate field values.
6. Press the PF4 function key to add the record.

### 7.36.4 Maintaining the Table

1. Display the TTS table access screen by typing /FOR VJTDSO.
2. In the TABLE NAME field, type "SOAC TO SOPID".
3. Leave the TABLE KEY blank, and type the desired SOACID in the TABLE RECORD KEY field.
4. Press the PF1 function key to find the desired record.
5. To update an entry, enter a C to the left of the field(s) being modified, make the appropriate modifications, and press the PF5 function key.

## 7.37 TCM SCHEDULE

### 7.37.1 Table Description

This table contains the TCM Trancodes and their associated Trancode Indicators. The indicators are used to prevent lockouts in the TCM SEC database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TCM SCHEDULE      TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: TCM SCHEDULE TABLE                         REL/LEV: 28.1      MOD: N
   FIELD     FIELD
   NAME      VALUE
   NBRTRAN   15*
   TRANCD1   VJMPCC1 *
   TRANCD2   VJMPCC2 *
   TRANCD3   VJMPCC3 *
   TRANCD4   VJMPCC1A*
   TRANCD5   VJMPCC1B*
   TRANCD6   VJMPCC1C*
   TRANCD7   VJMPCC1D*
   TRANCD8   VJMPD   *
   TRANCD9   VJMPM   *
   TRANCD10  VJMPCC2A*
   TRANCD11  VJMPCC2B*
   TRANCD12  VJMPCC2C*
   TRANCD13  VJMPCC1E*
   TRANCD14  VJMPCC1F*
TTS206I FIND COMPLETED  CONTINUED ON NEXT PAGE
```

**Figure 7–51.** TCM SCHEDULE Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TCM SCHEDULE      TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: TCM SCHEDULE TABLE                         REL/LEV: 28.1      MOD: N
   FIELD     FIELD
   NAME      VALUE
   TRANCD15 VJMPCC  *









TTS210I FIND COMPLETED
```

**Figure 7–52.** TCM SCHEDULE Table (2 of 2)

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore |
| **Updating Cautions:** | Should not be updated by anyone other than Bellcore. |
| **Associated Subsystem:** | TCM |
| **Module(s):** | VJMPFDT |

### 7.37.2 Field Descriptions

NBRTRAN

Full Name: Number of Transaction Codes

Definition: The number of transaction codes contained in the table.

TRANCD1 through TRANCD15

Full Name: Transaction code 1 through 15

Definition: The literal names of the transaction code indicators.

### 7.37.3 Creating the Table

This table is created and maintained by Bellcore.

### 7.37.4 Maintaining the Table

This table is created and maintained by Bellcore.

## 7.38 TCM USER CONTROL

### 7.38.1 Table Description

The TCM USER CONTROL table is used to identify the System Entity Code (SEC) of NSDB to its TCM communications module. The MYSEC field on the TCM USER CONTROL table contains the SEC associated with NSDB. Three other fields are also found on this table: TSALTERM, I/O BUF, and TPAM BUF. These fields are used to designate an LTERM for error messages (TSALTERM), allocate the I/O buffer value (I/O BUF), and allocate the buffer value for the TCM Parser and Mapper (TPAM BUF).

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TCM USER CONTROL  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0001
NOTE: TCM USER CONTROL TABLE                REL/LEV:            MOD: Y
   FIELD     FIELD
   NAME      VALUE
   MYSEC     VJIT1SEC*
   TSALTERM  PY511CG2*
   IN BUF    0500*
   OUT BUF   0500*
   #P BUF    0500*
   UNIV BUF  000*
   PRFON     N*
   PRFPGMS                                                          *
   PRFCMD1                                                          *
   PRFCMD2                                                          *
   PRFCMD3                                                          *
   PRFCMD4                                                          *
   PRFCMD5                                                          *
   PRFRSYS           *
   PRFTSYS           *
TTS206I FIND COMPLETED  CONTINUED ON NEXT PAGE
```

**Figure 7–53.** TCM USER CONTROL Table (1 of 2)

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TCM USER CONTROL  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: TCM USER CONTROL TABLE                     REL/LEV:          MOD: Y
   FIELD     FIELD
   NAME      VALUE
   PRFPATH           *
   PRFSCTYP  *
   PRFTMST                *
   PRFACNO                          *




TTS210I FIND COMPLETED
```

**Figure 7–54.** TCM USER CONTROL Table (2 0f 2)

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | MYSEC should always match a SEC entry in the TCM SEC Database (VJMPSCDD). Increase the I/O and TPAM buffer as experience indicates. |
| **Associated Subsystem:** | TCM |
| **Module(s):** | VJMPFDT |

### 7.38.2 Field Descriptions

MYSEC

Full Name: My (NSDB) System Entity Code

Definition: The literal value for NSDB's SEC.

Data Content: Up to 8 A/N

Example: VJIT1SEC

TSALTERM

Full Name: TCM System Administration Logical Terminal number

Definition: The LTERM where exception messages are to be sent.

Data Content: Up to 8 A/N

Example: PY511CG2

I/O BUF

Full Name: I/O Buffer size

Definition: The input / output buffer size allocation

Data Content: Up to 3 N

Example: 200

TPAM BUF

Full Name: TCM Parser and Mapper Buffer size

Definition: The TPAM buffer size.

Data Content: Up to 3 N

Example: 200

### 7.38.3 Creating the Table

To create the TCM USER CONTROL table, follow the steps below:

1. From a clear screen, type /FOR VJTDSO and press the ENTER key.
2. On the returned screen, type TCM USER CONTROL in the TABLE NAME field, and press PF1 to FIND.
3. On the returned screen, type in the SEC value assigned to NSDB in the MYSEC field.
4. Type in the LTERM number for TCM exception messages in the TSALTERM field.
5. Type in the buffer size allocations in the I/O BUF and TPAM BUF fields.
6. Press PF4 to ADD the values.

### 7.38.4 Maintaining the Table

The TCM USER CONTROL table requires very little maintenance. Only one TCM USER CONTROL table exists for NSDB. The SEC identity for NSDB in the MYSEC field should remain unchanged under normal conditions. The LTERM for exception messages can be changed to redirect messages if necessary. The buffer sizes can also be changed (increased, or decreased if necessary).

## 7.39 TIME ZONE OPTION

### 7.39.1 Table Description

From this table the user can select time zone processing options and define the host processor System Zone Name if this feature is activated. The Time Zone feature is used in NSDB for the first time with Release 3.2 in conjunction with the Measurement database. The measurement system supports time zone translations for the maintenance data date/time pairs sent from WFA/C.

When the TRANFLAG field has a value of "Y", NSDB uses the SYS ZONE field value in all date and time translations. The SYS ZONE field is a required entry to ensure proper implementation of the Time Zone feature within NSDB, and accurate system–to–system translation between NSDB and the WFA products.

Before turning on this feature, the default time zone name should have been established for each user. This is accomplished via the NSDB Security Administration screen (VJS1ADM). Each user can override his or her default time zone name by specifiying an effective time zone name on the VJS1SIGN screen when signing on. (Refer to the *NSDB Security Administrator's Guide* (BR 190–534–003).

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TIME ZONE OPTION  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                              # OF RECORDS: 0001
NOTE: TIME ZONE TRANSLATION OPTIONS           REL/LEV: 31.1.1.1  MOD: Y
   FIELD     FIELD
   NAME      VALUE
   TRANFLAG  N*
   SYS ZONE  EASTERN*
   DEBUG     N*









TTS210I FIND COMPLETED
```

**Figure 7–55.** TIME ZONE OPTION Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | None |
| **Table Type:** | DSECT |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore and BCC |
| **Updating Cautions:** | Before enabling the Time Zone feature (changing the translation flag from "N" to "Y"), check to be sure the ZONE NAMES TTS table is built. |
| **Associated Subsystem:** | SSC |
| **Related Table(s):** | ZONE NAMES |
| **Format(s):** | VJTZONE |
| **Run(s):** | VJRQR01 |
| **Module(s):** | VJTZDTX, VJTZONE |

### 7.39.2 Field Descriptions

TRANFLAG

| | |
|---|---|
| Full Name: | Translation Flag |
| Definition: | Indicates whether the Time Zone feature is turned ON (TRANFLAG = "Y") or OFF (TRANFLAG = "N" or blank). |
| Data Content: | 1 A |
| Valid Entries: | Y – Yes, perform Time Zone translation from the zone of the WFA/C sending system to the zone of the NSDB receiving system.<br><br>N or blank – No, do not perform Time Zone translations. |

SYS ZONE

| | |
|---|---|
| Full Name: | System Zone Name |
| Definition: | This field facilitates time zone processing with outside systems by defining a valid Zone Name for a system's processor. The value in this field is compared against ZONENAME in the ZONE NAMES TTS table to ensure that it is valid. |
| Data Content: | 7 A/N (any except the word "SYSTEM") |
| Valid Entries: | Valid entries include any Zone Name defined in the ZONE NAMES TTS table |

DEBUG

Full Name: Debug Processing Indicator

Definition: This field should only be used in conjunction with Bellcore support; it is associated with internal processing functions.

Data Content: To be supplied by Bellcore

Valid Entries: This field should be blank or populated with an "N" when viewed by a BCC user.

### 7.39.3 Creating/Maintaining the Table

1. Display the table by typing "/FOR VJTDSO".
2. Enter "TIME ZONE OPTION" in the TABLE NAME field.
3. Leave the TABLE KEY field blank.
4. Leave the TABLE RECORD KEY blank.
5. Press the FIND PF1 function key to find the desired data. If the record does not exist, continue with steps 6, 7, and 8. If the record exists, continue with steps 9 and 10 for updates and deletes, respectively.
6. Type an "A" (Add) to the left of the TRANFLAG field (under the "N" of "Note:") (optional step) and enter a value in the TRANFLAG field.
7. If the TRANFLAG value is "Y", enter a value in the SYS ZONE field.
8. Press the ADD PF4 function key to add the record.
9. To update an entry, type a "C" to the left of the field name(s) being modified, make appropriate modifications, and press the PF5 function key.
10. To delete a record (table), type a "D" to the left of the field name to be deleted (optional step) and press the DELETE PF10 function key.

# 7.40 TTS SECURITY

## 7.40.1 Table Description

This table is used to give permission to update a table by specifying the LTERM allowed to update, the Administrative Area that "owns" the table, and the primary subsystem(s) that are affected.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TTS SECURITY      TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0038
NOTE: TTS SECURITY NEED SUBSYS NAMES             REL/LEV:           MOD: Y
   FIELD    FIELD
   NAME     VALUE
   LTERM             *
   ADMINARA   *
   SUBSYS1           *
   SUBSYS2           *
   SUBSYS3           *
   SUBSYS4           *
   SUBSYS5           *
   SUBSYS6           *
   SUBSYS7           *
   SUBSYS8           *
   SUBSYS9           *
   SUBSYS10          *
   SUBSYS11          *
   SUBSYS12          *

TTS210I FIND COMPLETED
```

**Figure 7–56.** TTS SECURITY Table

| | |
|---|---|
| **Table Key(s):** | Refer to 190–534–003 |
| **Table Record Key(s):** | Refer to 190–534–003 |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | BCC |
| **Updating Cautions:** | Must be a valid subsystem from the TTS SUBSYSTEMS table. |
| **Associated Subsystem:** | TTS |
| **Related Table(s):** | TTS SUBSYSTEMS |
| **Module(s):** | VJTTSSVY |

### 7.40.2 Field Descriptions

LTERM

Full Name: Logical Terminal

Definition: The LTERM number.

Data Content: 8 A/N

Example: PY517CG1

ADMINARA

Full Name: Administrative Area Code

Definition: Identifies a group or geographic area having administrative responsibilities for TTS records listing that code.

Data Content: 2 A/N

Example: PS

SUBSYS1 through SUBSYS12

Full Name: Subsystem 1 through subsystem 12

Definition: Identifies the names of valid TTS subsystems.

Data Content: 12 fields, up to 8 A/N each.

Example: TTS
ODB

### 7.40.3 Creating the Table

See the *NSDB Security Administrator's Guide*, BR 190-534-003.

### 7.40.4 Maintaining the Table

See the *NSDB Security Administrator's Guide*, BR 190-534-003.

# 7.41 TTS SUBSYSTEMS

## 7.41.1 Table Description

This table contains validation records that identify the primary sybsystems for which security tables are created. See the *NSDB Security Administrator's Guide*, BR 190-534-003.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TTS SUBSYSTEMS     TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                          # OF RECORDS: 0012
NOTE: TABLE OF SUBSYSTEMS VALID TO TTS    REL/LEV: 2.1       MOD: N
   FIELD     FIELD
   NAME      VALUE
   SUBSYS            *









TTS210I FIND COMPLETED
```

**Figure 7–57.** TTS SUBSYSTEMS Table

| | |
|---|---|
| **Table Key(s):** | Refer to 190–534–003 |
| **Table Record Key(s):** | Refer to 190–534–003 |
| **Table Type:** | Validate |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore and BCC |
| **Updating Cautions:** | Must have valid subsystem names as identified in the table descriptors for each table created. There must be one blank record to accommodate possible blank Subsystem 1 to Subsystem 12 fields in the TTS SECURITY table. |
| **Associated Subsystem:** | TTS |
| **Related Table(s):** | TTS SECURITY |
| **Format(s):** | VJTDSO, VJTBDO |
| **Module(s):** | VJTTBLRV |

### 7.41.2 Field Descriptions

SUBSYS

Full Name: Subsystem

Definition: Identifies the name of a valid TTS subsystem.

Data Content: 8 A/N

Valid Entries: Any valid TTS subsystem

### 7.41.3 Creating the Table

See the *NSDB Security Administrator's Guide*, BR 190-534-003.

### 7.41.4 Maintaining the Table

See the *NSDB Security Administrator's Guide*, BR 190-534-003.

## 7.42 TTS VALID XREF

### 7.42.1 Table Description

The TTS VALID XREF table is used primarily by Bellcore to set up validations for TTS tables. The tables defined as Source Table (SOUTABLE) and Destination Table (DSTTABLE) must each carry the field specified in the Source Field (SOUFIELD), and this field must carry the same value in both tables.

The TTS VALIF XREF table is shipped as a locked table by Bellcore, prepopulated with data. It should contain a non–zero number of entries (OQS will receive a time zone translation error if this table is not properly defined). In Release 3.2, it is used to support time zone operations associated with the Measurement database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  TTS VALID XREF     TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0005
NOTE: TTS VALIDATION CROSS REFERENCE TABLE      REL/LEV: 4.1.1.4   MOD: N
   FIELD     FIELD
   NAME      VALUE
   SOUTABLE                  *
   SOUFIELD          *
   OPTIONAL  *
   DEFAULT                       *
   DSTTABLE                  *
   VALTYPE   *
```

**Figure 7–58.** TTS VALID XREF Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | |
| **Table Type:** | |
| **Initial Issue:** | |
| **Data Supplied by:** | Bellcore |
| **Updating Cautions:** | None – Prepopulated by Bellcore |
| **Associated Subsystem:** | TTS |

### 7.42.2 Field Descriptions

SOUTABLE

Full Name: Source Table

Definition: The table to be used as the source for the field defined in SOUFIELD.

Data Content: 16 A

Valid Entries: A valid TTS table name

SOUFIELD

Full Name: Source Field

Definition: The field specified as a Source Field is compared in the tables identified by SOUTABLE and DSTTABLE.

Data Content: 8 A

Valid Entries: A valid TTS field name

OPTIONAL

Full Name: Optional

Definition: Controls the activity of the validation module.

Data Content: 1 A

Example: M (show Message for field)

DEFAULT

Full Name: Default

Definition: This field is not currently used.

DSTTABLE

Full Name: Destination Table

Definition: The table to be used as a destination for the field defined in SOUFIELD.

Data Content: 16 A

Valid Entries: A valid TTS table name

VALTYPE

Full Name: Validation Type

Definition: This field is not currently used.

### 7.42.3 Creating the Table

The TTS VALID XREF table is created and maintained by Bellcore. It is prepopulated with data and shipped to the BCCs.

# 7.43 ZONE NAMES

## 7.43.1 Table Description

This table contains fields (Zone Name, Winter ID, and Summer ID) that are needed to support the NSDB SYS ZONE that is used in the TIME ZONE OPTION TTS table. The table record key is the Zone Name which maps to fields for winter and summer Zone IDs (e.g., ADT, AST, CDT, CST, EDT, EST, MDT, MST, PDT, PST). Zone IDs are validated against the ZONE OFFSETS TTS table.

Record information for this table is developed and entered by Bellcore. This table is used for the first time in Release 3.2 to support time zone operations associated with the Measurement database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ZONE NAMES        TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0001
NOTE: TIME ZONE NAMES AND THEIR ZONE ID'S      REL/LEV:            MOD: Y
   FIELD     FIELD
   NAME      VALUE
   ZONENAME          *
   WINTERID      *
   SUMMERID      *
```

**Figure 7–59.** ZONE NAMES Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Valid Zone Name |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore |
| **Updating Cautions:** | There must be an entry for the system zone. The zone IDs must exist in the ZONE OFFSETS TTS table. |
| **Associated Subsystem:** | SSC |
| Related Table(s): | TIME ZONE OPTION,<br>ZONE OFFSETS |
| **Format(s):** | VJTZONE |
| **Run(s):** | VJRQR01 |
| **Module(s):** | VJTZDTX, VJTZONE, VJTZVLZN |

### 7.43.2 Field Descriptions

ZONENAME

| | |
|---|---|
| Full Name: | Zone Name |
| Definition: | The name for a specific time zone. |
| Data Content | 7 A/N |
| Valid Entries: | Any combination of up to seven characters |

WINTERID

Full Name: Winter Zone ID

Definition: Indicates the Zone ID applicable during the winter time period. The zone ID is validated against the ZONE OFFSETS TTS table.

Data Content: 3 A

Valid Entries:
ADT – Atlantic Daylight Time
AST – Atlantic Standard Time
CDT – Central Daylight Time
CST – Central Standard Time
EDT– Eastern Daylight Time
EST – Eastern Standard Time
MDT – Mountain Daylight Time
MST – Mountain Standard Time
PDT – Pacific Daylight Time
PST – Pacific Standard Time

SUMMERID

Full Name: Summer Zone ID

Definition: Indicates the Zone ID applicable during the summer time period. The zone ID is validated against the ZONE OFFSETS TTS table.

Data Content: 3 A

Valid Entries:
ADT – Atlantic Daylight Time
AST – Atlantic Standard Time
CDT – Central Daylight Time
CST – Central Standard Time
EDT– Eastern Daylight Time
EST – Eastern Standard Time
MDT – Mountain Daylight Time
MST – Mountain Standard Time
PDT – Pacific Daylight Time
PST – Pacific Standard Time

### 7.43.3 Displaying the Table

1. Display the table by typing "/FOR VJTDSO".
2. Enter "ZONE NAMES" in the TABLE NAME field.
3. Leave the TABLE KEY field blank.
4. Enter the Zone Name in the TABLE RECORD KEY field.
5. Press the FIND PF1 function key to find the desired data.

## 7.44 ZONE OFFSETS

### 7.44.1 Table Description

The purpose of this table is to map the Zone ID to an offset of Greenwich Mean Time (GMTOFST). The table record key is the Zone ID (e.g., MST, MDT, EST, EDT, etc.) with an associated Greenwich Mean Time offset value. The GMT offset value is displayed in minutes. This value is the offset of the Zone ID in minutes (plus or minus) from Greenwich Mean Time. Standard Zone IDs (EST, MDT, etc.) for the United States are included in the table.

The records cannot be edited by the user. Record information for this table is developed and entered by Bellcore. The ZONE OFFSETS table is used for the first time in Release 3.2 to support time zone operations associated with the Measurement database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  ZONE OFFSETS      TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS: 0010
NOTE: ZONE ID AND CORRESPONDING GMT OFFSET      REL/LEV: 31.1.1.1  MOD: N
   FIELD     FIELD
   NAME      VALUE
   ZONEID      *
   GMTOFST     *
```

**Figure 7–60.** ZONE OFFSETS Table

| | |
|---|---|
| **Table Key(s):** | None |
| **Table Record Key(s):** | Time Zone IDs |
| **Table Type:** | Passback |
| **Initial Issue:** | 3.0 |
| **Data Supplied by:** | Bellcore (locked table) |
| **Updating Cautions:** | None |
| **Associated Subsystem:** | SSC |
| **Related Table(s):** | ZONE NAMES |
| **Format(s):** | VJTZONE |
| **Run(s):** | VJRQR01 |
| **Module(s):** | VJTZDTX, VJTZONE |

### 7.44.2 Field Descriptions

ZONEID

| | |
|---|---|
| Full Name: | Time Zone Identification |
| Definition: | The time zone identification for a geographic region where the same standard time is used. |
| Data Content | 3 A |
| Valid Entries | ADT – Atlantic Daylight Time<br>AST – Atlantic Standard Time<br>CDT – Central Daylight Time<br>CST – Central Standard Time<br>EDT– Eastern Daylight Time<br>EST – Eastern Standard Time<br>MDT – Mountain Daylight Time<br>MST – Mountain Standard Time<br>PDT – Pacific Daylight Time<br>PST – Pacific Standard Time |

GMTOFST

Full Name: Greenwich Mean Time (GMT) Offset

Definition: The offset of the Zone ID in minutes (plus or minus) from Greenwich Mean Time.

Data Content: 4 (1 = + or –, then 3N)

Valid Entries: The + or – symbol and any numeric combination

### 7.44.3 Displaying the Table

1. Display the table by typing "/FOR VJTDSO".
2. Enter "ZONE OFFSETS" in the TABLE NAME field.
3. Leave the TABLE KEY field blank.
4. Enter the Zone ID in the TABLE RECORD KEY field.
5. Press the FIND PF1 function key to find the desired data.

## Recommended Runs and Their Usage
## CONTENTS

# 8. Recommended Runs and Their Usage

This section describes some of the runs that are used in ongoing NSDB operations. The purpose of this section is to give the NSDB system administrator a general understanding of the most commonly–used runs to facilitate coordination with the ISO group.

The runs in this section are described in greater detail in the runbooks provided by Bellcore. A complete listing can be obtained by reviewing the NSDB Product List, which should be available through your Documentation Coordinator.

Some of the runs described in this section require the user to supply transaction codes. These codes are described in the Bellcore document BR 190-534-002, the *NSDB Format To Transaction Cross Reference Guide.* This restricted document is provided to the BCCs on a one-per-region basis, and can be obtained through the Bellcore Technical Support Center (TSC).

To understand the run descriptions, you will need to understand the following terms:

- Run

  A Run is a procedure used for a mainframe computer to perform large–scale maintenance activities, extraction of desired data from a database, loading of data into a database, and updates to a database.

- Batch Run

  A Batch Run is a type of run that is performed only under certain conditions. "Pure" Batch conditions require the run to be performed outside of IMS control while the application (i.e., database) is unavailable to users.

- BMP Run

  A Batch Message Program (BMP) Run is a type of run that is initiated and scheduled by IMS (the Operating System Software). A BMP Run processes groups of messages that access online databases. The characteristics of BMP Runs as compared to Batch Runs are:

  - BMP Runs are performed under the control of IMS
  - BMP Runs require that the databases are up and running
  - Batch Runs are performed outside the control of IMS
  - Batch Runs are performed while the databases are off-line.

- DLI

  A DLI Run is a type of run that can be performed with or without the control of IMS. The databases must be off-line for a DLI Run.

- Control Card

  A Control Card is used to select various options available with a Run. Typical options include performance options, extract, load, and update criteria selections, and Run mode. (The mode selection allows the Run to be performed on a read or update basis. The read mode allows you to see what would happen as a result of an update Run, without actually making the updates.)

  There are two kinds of Control Cards. The first kind is known as instream JCL. This kind of Control Card has user–selectable options that influence the Run and the resultant data. This is the type of Control Card for which the NSDB system administrator must supply input in order for the ISO group to initiate the run.

  The second kind of Control Card is known as "fixdata". This type of Control Card does not have user–selectable options; it is shipped from Bellcore prepopulated.

The run type can be determined by examining the fifth character of the run identification name. The list of valid run types follows:

C – Creation/Conversion

I – Initialization

L – Load

M – Merge or Maintenance

P – Purge

U – Utility

R – Report

The following run types are associated with TIRKS and CORE for a subsystem of installation:

D – CL

E – EL

Q – QOC

S – Span

X – Cable

## 8.1 VJARM01 – NSDB Archive Run

*Run Description*

This run archives records from various databases as identified by an Archive Name that is supplied as a parameter to the PROC. The actual records to be archived are controlled by data entered through the online screens VJARCNTL and VJARDSPA. Information about age criteria is entered on VJARCNTL as a three–character field. The VJARDSPA screen is used to control the use of a backup tape and, if needed, additional field–level selection criteria.

After a record has been archived, it is deleted from the database. An option exists to write the data to tape and not delete it from the database. In NSDB, conversely, an option also exists to delete the data from the database and not write it to tape.

In addition to the archived data, this run produces an index file which contains record keys and record pointers to provide for quick retrieval of the data at a later time. The DSN for the archive tape is generated within the code using the HLQ and ARCHNM symbolic parameters from the JCL, and the date and time when the tape is first used.

Valid Archive Names for NSDB are:

| Archive Name | Database |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

VJARM01 is executed in BMP mode.

For more information about the VJARM01 run, refer to BR 190–534–058.

*Recommendations*

Run Frequency – On request

Prerequisite(s) – Proper archive selection criteria should be entered/verified via online screens VJARCNTL and VJARDSPA.

Indices for Generation Data Groups must be created initially for STEP05.SYSUT2. One index must be created for each valid value of the Archive Name. It is suggested that LIMIT=10 be used when creating the GDG indices.

Postrequisite(s) – None

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP03.PGMCNTL
Number of steps: 5

## 8.2 VJARM02 – NSDB Archived Data Retrieval Run

*Run Description*

This run is used to retrieve data from tapes that were created by using the archive run VJARM01. It uses selection criteria from the online screen VJARDSPR and the parameters supplied to the PROC. These parameters and selection criteria are used to scan the index file associated with the archive named in the parameters and to identify tapes and records on the tapes that will be retrieved.

The scan of the index file can be done either as a part of this run or in advance using the run VJARM03. If VJARM03 is used, a list of tapes and records is obtained from the database VJARSLDP. In either case, the data is read from tape and any additional selection criteria is applied. If the record passes the selection criteria, the record is printed in a generic format (FIELD_NAME DESCRIPTION VALUE).

Valid Archive Names for NSDB are:

| Archive Name | Database |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

VJARM02 is executed in BMP mode.

For more information about the VJARM02 run, refer to BR 190–534–059.

*Recommendations*

Run Frequency – On request

Prerequisite(s) – Proper retrieval selection criteria should be entered/verified via the online screen VJARDSPR.

Postrequisite(s) – None

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP03.PGMCNTL
Number of steps: 5

## 8.3 VJARM03 – NSDB Tape Identification Process

*Run Description*

This run is used to identify which tapes will be needed for a retrieval in advance of the actual retrieval run. It uses the selection criteria from online screen VJARDSPR and the parameters supplied to the PROC. These parameters and the selection criteria are used to scan the index file associated with the archive named in the parameters, and to identify tapes and records on the tapes that would be retrieved if a retrieval were run with the same criteria. This data is saved in the selection database and passed to the next retrieval run. The count of records and list of tapes can be reviewed from the online screens VJARLSTR and VJARTLST.

If a retrieval run is made before the next archive run or changes are made to the selection criteria, the work performed during this run will be passed directly to the retrieval run. If, however, the user changes the selection criteria before the retrieval run, the saved information is considered out–of–date and is deleted. If a new archive run is made before the retrieval run, this information is also considered out–of–date, and the retrieval run will recalculate the tape information.

Valid Archive Names for NSDB are:

| Archive Name | Database |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

VJARM03 is executed in BMP mode.

For more information about the VJARM03 run, refer to BR 190–534–060.

*Recommendations*

Run Frequency – On request

Prerequisite(s) – Proper retrieval selection criteria should be entered/verified via the online screen VJARDSPR.

Postrequisite(s) – None

Subsequent Run(s) – If the output from this run is as expected, the actual data retrieval is performed by run VJARM02.

*Control Card Information*

User control cards: STEP03.PGMCNTL
Number of steps: 4

## 8.4 VJCDE02 – NSDB Measurements Database Extract

*Run Description*

This program executes SQL statements to extract data from the NSDB Measurement database. It produces one extract file for maintenance information, and one for installation information. Users can configure the VJCDE02 to extract only maintenance tables or installation tables, as desired.

The extract files will allow BCC programmers to generate record layouts locally in COBOL, PL1, or Assembler. VJCDE02 will append a two–character record type to the beginning of each extracted record, and will sort all extract files into one output file, grouped by trouble report number.

VJCDE02 is executed in batch mode under the TSO Monitor Program (IKJEFT01).

For more information about the VJCDE02 run, refer to BR 190–534–092.

*Recommendations*

Run Frequency – As needed.

Prerequisite(s) – The plan (DB2 extract program) VJCD02E must be bound to the DB2 catalog by the TSO user ID which will be the owner of all Measurement database objects.

Postrequisite(s) – None

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP01.SYSIN and STEP03.SYSIN
Number of steps: 4

## 8.5 VJCDL01 – Bulk Load for NSDB Databases

*Run Description*

This run is used to load the NSDB Circuit/Carrier and Message databases using a TIRKS system extract produced by run ZRCDX07. VJCDL01 can also be used to load the TIRKS system Index Page data for multi-point circuits, as well as to load Special Service circuit data into the NSDB Line Record and Alias databases.

With NSDB Release 3.2, the VJCDL01 simultaneously loads the Facility–to–Circuit tables when the FACCKTDB flag in the ODB CO OPTIONS TTS table and the fields in the ODB FAC CKT DB TTS table are set to "Y". Specifically, the VJCDL01 loads in–effect special circuits, message circuits, and carrier circuits into the Interoffice Cable to Circuit and Carrier to Circuit tables.

NMA/Switch uses this run to get TIRKS data into the NMA database. Circuits needed by the NMA/Switch system are extracted from the TIRKS system (ZRCDX07), written to a file (VJCDL01), and then converted to Flexible Computer Interface Format (FCIF) by another NSDB run, VJCDU04. NMA load software loads the reformatted records into the NMA database.

VJCDL01 is executed as a BMP run.

*Recommendations*

This run is part of the NSDB load process for specials, carrier, and message circuits from TIRKS.

*Control Card Information*

Control card options for this run are limited to performance options (checkpoint and synchronization), and an option to replace (overlay) NSDB data with data found in the TIRKS extract.

## 8.6 VJCDL09 – Extract NSDB Circuit and Facility Data

*Run Description*

The purpose of this run is to perform an extract of the cable and carrier data for in–effect circuits to be loaded into the Facility–to–Circuit tables in postrequisite jobs. With the NSDB 3.2 release, there are two tables: the Interoffice Cable to Circuit table (VJDFCITT) and the Carrier to Circuit table (VJDFCXTT).

This job results in two load files (one for each table) along with an X05–type file containing those circuits with data prohibiting them from being loaded to the load files. This data can be reloaded from the TIRKS system using jobs contained in the "Subsequent Runs" section of the runbook. Also created is the partitioning element value which is used in the DDL for these tables.

This job is a useful tool for recreating either the VJDFCITT or VJDFCXTT table (see control card CTRLPRM) in the event of a data problem requiring extensive corrective work. It may be faster to simply delete all entries in the problem table and reload it using this run.

VJCDL09 can be executed in either DLI or BMP.

For more information about the VJCDL09 run, refer to BR 190–534–091.

*Recommendations*

Run Frequency – this process is to be used as the initial load for Facility–to–Circuit tables. It is run only once unless a problem occurs which requires the tables to be loaded again from scratch. In the event of such a problem, all entries in the tables must be deleted before a reload.

Prerequisite(s) – no prerequisite runs are required or optional unless the tables need to be deleted and loaded from scratch as a result of unanticipated problems.

Postrequisite(s) – the DB2 load utility must be run after the VJCDL09 in order to physically load the Interoffice Cable to Circuit table and the Carrier to Circuit table. This requires two runs of the IBM DB2 utility DSNUTILB. Each run will load one table. For each execution it is necessary to specify a SYSREC override which contains the DB2 load cards as well as a SYSIN file which contains the actual data from this job output. The following DDL members are used to load the tables from the corresponding load files:

Load the Interoffice Cable to Circuit table:
SYSREC: VJDFCILT
SYSIN: VJCD01GS output file (&DBNODE..CABLE.LOAD.VJCDL09)

Load the Carrier to Circuit table:
SYSREC: VJDFCXLT
SYSIN: VJCD02GS output file (&DBNODE..CARRIER.LOAD.VJCDL09)

Subsequent Run(s) – After the completion of this run, it is *optional* to run the ZRCDX07 job to re–extract the circuits from the TIRKS system which were output on the VJCD03GS gsam file (X05–type). This file was produced to allow circuits with errors on the cable and carrier entity segments to be reloaded into NSDB from TIRKS if desired. The process involves running the file through ZRCDX07, and then continuing through the flow to VJCDM01 and VJCDL01. Refer to the following runbooks for details about these jobs:

| Run ID: | BR#: |
|---|---|
| ZRCDX07 | BR 756–535–512 |
| VJCDM01 | BR 190–534–504 |
| VJCDL01 | BR 190–534–501 |

*Control Card Information*

User control cards: STEP01.CTRLPRM and STEP01.PGMCNTL
Number of steps: 3

## 8.7 VJCDM12 – Create Load File for Hunt Group Database

*Run Description*

The VJCDM12 run reads in access files that contain Hunt Group records, and it processes them to create the OCTL, HUNT, and TRAG aggregates. These aggregates are sorted in STEP02 to put them in the correct order (i.e., OCTL, HUNT, and TRAG). The sorted aggregates become input to the Generic NSDB Load module, VJCDL04. This load processes only in–effect records. VJCDM12 is not an IMS job.

VJCDM12 is executed in batch mode.

For more information about the VJCDM12 run, refer to BR 190–534–079.

*Recommendations*

Run Frequency – VJCDM12 should be run when NSDB 3.1.1 is installed if there are Hunt Group records in the Line Record and/or Set Services databases. It should be run later whenever there are loads into the Line Record and Set Services databases that contain Hunt Group records. The LMOS and ISDN loads into the Line Record database are examples of loads that may contain Hunt Group records.

Prerequisite(s) – VJCDX03 and VJCDX04

Postrequisite(s) – VJCDL04

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP01.SYSIN
Number of steps: 2

## 8.8 VJCDM13 – Population of the COTYPE Field on the ODM EXK TO SW ID TTS Table

*Run Description*

The VJCDM13 run is instrumental in populating the new COTYPE field on the ODM EXK TO SW ID TTS table in support of the Measurement database for NSDB Release 3.2. It reads the NSDB 3.1.x version of the flat file (from the VJCSM04 run) and user–entered input to produce an output reload file for the NSDB 3.2 version of the ODM EXK TO SW ID table.

VJCDM13 is executed as a DLI run.

For more information about the VJCDM13 run, refer to BR 190–534–103.

*Recommendations*

Run Frequency – As required.

Prerequisite(s) –

- The VJCSM04 run unloads the NSDB 3.1.x version of the ODM EXK TO SW ID table into a flat file in preparation for a conversion to an NSDB Release 3.2 level. It is designed to be used only once, as a part of the conversion process.
- The VJCSC08 run updates descriptors to include the new COTYPE field associated with the Measurement database.

Postrequisite(s) – None

Subsequent Run(s) – The VJCSC08 run is used again to reload the ODM EXK TO SW ID table.

*Control Card Information*

User control cards: STEP01.TYPEFLE, STEP01.EXKSWID, STEP01.TBLSEQ, and STEP01. PARMSIN

Number of steps: 1

## 8.9 VJCDM48 – NSDB Partial Location Reference Synchronization

*Run Description*

The VJCDM48 run answers the need to make location reference data from the TIRKS system available to other systems. This run permits a smooth flow of data, and does not cause exception processing at the occurrence of missing or erroneous location reference data.

VJCDM48 is a two–step run. The input is a file of location reference data produced by the TIRKS run ZRCDU06, DDNAME=ZRCD02GS. This file is sorted in STEP01 of this run. The sort causes the locations in the file to fall within three groups: free–standing locations, sublocations, and master locations, in that order. The locations are in alphabetical order within each of these subgroups. The second step of VJCDM48 processes the sorted file and updates the target location reference database.

VJCDM48 is executed either as a DLI or BMP run.

For more information about the VJCDM48 run, refer to BR 190–535–018.

*Recommendations*

Run Frequency – This run may be executed as needed to bring TIRKS location reference data into any system having a Location Reference database. It must be executed prior to running VJCDU21, the NSDB Master Location Propagation.

Prerequisite(s) – ZRCDU06 (BR 756–535–021)
TIRKS Extract Location Information

Postrequisite(s) – None

Subsequent Run(s) – VJCDU21 (BR 190–534–069)
NSDB Master Location Propagation

*Control Card Information*

User control cards: STEP01.SYSIN, STEP02.PGMCNTL, and STEP02.PARMSIN
Number of steps: 2

## 8.10 VJCDU16 – LFACS (LEIS) Loop Data Extract & Reformat

*Run Description*

This run is used to load NSDB with local loop data from LFACS. The process requires input from an LFACS/LEIS extract tape. VJCDU16 takes the LFACS data and produces NSDB data aggregates that can be used as input to VJCDL04, the NSDB Contract Facilitator run.

Updates are applied to existing circuits in NSDB. LFACS data for circuits that are not existing in NSDB is ignored. The updates are finally applied by VJCDL04 using the RCVWO (Receive Work Order) contract process.

*Recommendations*

The LFACS/LEIS extract tape used as input for VJCDU16 requires special care during data transfer, and the inputs should not be concatenated. See runbook 190-534-056 for details.

*Control Card Information*

No user supplied control card input is required for this run.

## 8.11 VJCDU17 – NSDB Bulk Update

*Run Description*

The purpose of this run is to enable bulk updates to selected fields in the Circuit/Carrier, Message, Multi-Point, and Line Record databases in NSDB. The companion run, VJCDU20, applies the updates generated by VJCDU17 to the NSDB databases. Input for the NSDB Bulk Update run can come from NSDB OQS queries (via VJOQR04 output subfiles), or from an external system providing FCIF input. When OQS subfiles are used as input for VJOQR04, the OQS data is sorted, validated, and reformatted. Reformatting is also required for external system FCIF files used as input to VJCDU17.

The following table shows the NSDB data fields that can be bulk updated, and their corresponding fields in WFA/C, TIRKS, and NMA.

| CORRESPONDENT BULK UPDATABLE FIELDS ACROSS SYSTEMS | | | |
|---|---|---|---|
| | | | |
| **NSDB** | **WFA/C** | **TIRKS** | **NMA** |
| | | | |
| ACNA | ACNACUST | ACNA | IC-NAME |
| BTN | BILL | BTN | |
| CUSTNAME | CNAME | CUST | |
| MCO | MCO | MCO | MCO |
| LPDATAA_LPCONTNAME | P1NAME | LCON NAME A | |
| LPDATAA_LPSTAADRS | P1ADDRESS | STA ADDR A | |
| LPDATAZ_LPCONTNAME | P2NAME | LCON NAME Z | |
| LPDATAZ_LPSTAADRS | P2ADDRESS | STA ADDR Z | |
| RRI | RRI | RRI | |
| TESTIND | RTI | RTI | |
| OCO | OCO | OCO | |
| CCO | CCO | CCO | |
| MCN | MCN | MCN | MCN |
| TSP | TSP | TSP | TSP |
| ESV | ESV | CESV | |
| AVN | AVN | CAVN | |
| ES | ENH_SERVICE | | |
| CLSV | CLSV | | |
| MCTR | MSSC | | |
| CUSCODE | CUSCODE | CUSCODE | |
| CUSADDR | CUSADDR | CUSADDR | |
| CCNA | CCNA | CCNA | |
| SN | | | |

The following NSDB data fields are validated before update is allowed:

| | |
|---|---|
| MCO | Validated against LOCREF TTS table |
| RRI | For the hours value (HH), valid values are 00–23. For the minutes value (MM), valid values are 00–59. |
| TESTIND | Valid values are Y, N, or U |
| OCO | Validated against LOCREF TTS table |
| CCO | Validated against LOCREF TTS table |
| TSP | First character can be the letter E or numbers 0 – 5; the second character can be number 0 – 5. |
| ES | Valid values are MP, IP, EP, or CO. |
| CLSV | Validated against NSDB Fixdata member. |
| MCTR | Validated against LOCREF TTS table |

*Recommendations*

The subfile output of VJOQR04 should be used when NSDB OQS queries are used as input to VJCDU17. A postrequisite run, VJCDU20, is also required to update the NSDB databases.

*Control Card Information*

The following user entered control card information is required:

- STEP01.PARMSIN

  The type of input used for the run is identified here.

  FILETYPE = 'OQS' (For OQS query subfile input)
  FILETYPE = 'FCIF' (For FCIF files from external systems)

- STEP03.PGMCNTL.

  The Overlay Rules are identified here.

  **OVERLAY BLANK**

  If the database value is blank, replace it with the new field value. If the database value is not blank, do not change the database value.

  **OVERLAY ALL**

  Always overlay the value in the database.

  **OVERLAY NONBLANK**

  Replace database values as long as the new field value is not blank (i.e., avoid overlaying values with a blank).

Sample Control Card entries for STEP03.PARMSIN

```
OVERLAY NONBLANK TSP
OVERLAY ALL AVN
OVERLAY BLANK MCO
```

The example above would replace the TSP values for those records where the replacement value is not a blank,

and

replaces every AVN value with the update AVN value,

and

replaces all blank MCO values with the update MCO values.

## 8.12 VJCDU21 – NSDB Master Location Propagation

*Run Description*

The VJCDU21 run propagates Master location LPC codes from the Location Reference database or converts from Sub to Master locations in the NSDB Circuit Carrier, Message, and Circuit Index databases. VJCDU21 is required for message and switched special service circuits that originate and/or terminate at a central office switch. Since the switch may be subdivided into more than one Sublocation, this job uses the Sublocation that was originally loaded into NSDB from the TIRKS/GOC system. The NMA system, however, is knowledgeable about the switch's Master location and uses it to communicate troubles to WFA/C. TIRKS Release 16.0.1 will begin passing Master location information to NSDB and the NMA system. VJCDU21 updates the message circuits and special circuits loaded before this TIRKS release.

VJCDU21 sequentially scans the NSDB In–Effect and Pending Circuit Carrier and Message databases for circuits with trunk translation data. The NSDB Location Reference database will be accessed to determine if a location is a Master or a Sub, and it is re–accessed to get the Sub's Master when it is appropriate. A flag in the Circuit Header will be set to "M" (Master) when the trunk translation location is already a Master or when it is being updated to a Master. The LOC field in the trunk translation data will be updated to contain the Master when it is appropriate. This procedure will be initiated for both the A end and the Z end of the circuit.

Control cards allow the VJCDU21 to be run for all of or a specified subset of the partitions for the NSDB Circuit Carrier and Message databases.

The Circuit Index of each circuit with trunk translation data will be retrieved. If the LPC code in the tag–keys tag/value format data contain the Sublocation, it will be updated to the Master LPC. If no tag–keys are found for a given circuit, it will be written to a CAC/CLO X05–type file for future extraction from the TIRKS system.

Any inconsistencies with the Location Reference Master/Sub relationships will be reported on an Error report. An optional Detailed report of all updates will also be provided.

INPUT:

Control Cards

OUTPUT:

Updated NSDB databases
Optional Detailed report
Statistics and Error report
X05–type CAC/CLO file of circuits with bad tag–keys.

**NOTE** – This job can be gracefully brought down while running in BMP mode. The command from an IMS terminal is "*VJCD21U1 SHUT DOWN*". VJCD21U1 is the transaction code. The job can be checkpoint – restarted after bringing it down in this fashion.

**NOTE** – Multiple executions of this run can be done in parallel by using the following procedures:

1. Preferrably, request a different database or, at least, different database partitions via "UPDTOPT" control cards.
2. Use a separate transaction ID for each run. Use VJCD21U1 and the available spare BMP transaction ID, VJCDSP16.
3. Use different job names in the job card for each run.
4. Use different database names for the VJCD01GS output GSAM file generated by the job.

VJCDU21 is executed either as a DLI or BMP run.

For more information about the VJCDU21 run, refer to BR 190–534–069.

*Recommendations*

Run Frequency – One time, as long as it is run after TIRKS Release 16.0.1 has been installed

Prerequisite(s) – VJCDM48 (BR 190–535–018)
NSDB Partial Location Reference Synchronization

Postrequisite(s) – None

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP01.PGMCNTL, STEP01.UPDOPTS
Number of steps: 1

## 8.13 VJCDU22 – NSDB Segment Version Update

*Run Description*

Due to the NSDB Dynamic Conversion process available with NSDB Release 3.1, it is possible that databases might contain data from a variety of releases. This does not pose a problem for the online processes, but the disparity in data formats affects many DASD batch runs and OQS–related queries. The VJCDU22 run has been designed to synchronize this data in consistent formats (the latest versions) at the outset of a new release, and it has the potential to include all future contract/function combinations supported by NSDB.

The VJCDU22 BMP run is the first in a sequence which invokes the NSDB Supervisor to perform certain NSDB functions (particularly the LRVRSN–version update for the Line Record database and REBUILD).

Input for the VJCDU22 run consists of user–supplied control cards and names of the databases necessary to process them. Input may specify particular databases for a desired function. The VJCDU22 process obtains the keys for the records associated with the function, and produces an output file of the aggregates with this key information. This file of aggregates is then used as input for the VJCDL04 run (the NSDB Contract Facilitator). The VJCDL04 causes the functions requested in the VJCDU22 to be performed.

The LRVRSN function is required for NSDB Release 3.1.1. This function causes the Line Record database to be updated (all records will be remapped to the latest versions of the formats).

VJCDU22 is executed as a BMP run.

For more information about the VJCDU22 run, refer to BR 190–534–081.

*Recommendations*

Run Frequency – The version function for this run should be performed at the outset of any new release which contains database changes that are the result of online conversions. It should be performed against all appropriate databases. Bellcore will contact the BCCs when this job is necessary. All other functions are run on an as–needed basis.

Prerequisite(s) – None

Postrequisite(s) – None

Subsequent Run(s) – It will be necessary to run the VJCDL04 (BR 190–534–052) after the VJCDU22 to perform the functions requested in the VJCDU22. The output file from the VJCDU22 in the dd statement (VJCD01GS) will be specified in the execjcl for the VJCDL04 dd statement (also VJCD01GS).

*Control Card Information*

Control parameters contain data to direct the program to the database(s) to be processed, the level status of the records to be processed, the contract to invoke, and the function to be passed to the VJCDL04 job.

User control cards: STEP01.CTRLPRM and STEP01.PGMCNTL
Number of steps: 1

## 8.14 VJCDU23 – Create FCIF Output from Tag/Value Input

*Run Description*

The VJCDU23 run simplifies the BCC's migration of data into NSDB databases by automating the construction of records into an FCIF format that is acceptable to the VJCDM09 run.

There is still a need to have an understanding of the FCIF aggregate hierarchy and a Bellcore–supplied description of what data should be in which aggregates. The specific hierarchy to use is defined by Bellcore in a loadable CSECT known as a Message Format Descriptor or "MFD".

VJCDU23 is executed as a Batch run.

For more information about the VJCDU23 run, refer to BR 190–534–082.

*Recommendations*

Run Frequency – As required when loading data into NSDB.

Prerequisite(s) – User–written procedure to produce this job's input dataset.

Postrequisite(s) – VJCDM09 (BR 190–534–036) FCIF Merge to MDA Formatter.

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP01.SYSIN
Number of steps: 1

## 8.15 VJCDU28 – Measurement Database Load – Maintenance

*Run Description*

This run is a part of the Measurement Database feature. It uses formatted output from the WFA/C extract run, VOCDU28, to load the Measurement database DB2 tables.

VJCDU28 is executed as a BMP run.

For more information about the VJCDU28 run, refer to BR 190–534–095.

*Recommendations*

Run Frequency – As needed. This run can be used to establish a "base" for the Measurement database after conversion to NSDB 3.2 and WFA/C 1.2. If this run is used as a part of the batch measurements interface, it should be executed at regular intervals to keep the measurement database updated.

Prerequisite(s) – VOCDU28 (WFA/C Measurement Database Extract – Maintenance)

Postrequisite(s) – None

Subsequent Run(s) – After the first load, or if a large number of records are loaded, the DB2 RUNSTATS utility should be run to update the DB2 catalog.

*Control Card Information*

User control cards: STEP01.PGMCNTL and STEP01.IMSCNTL
Number of steps: 1

## 8.16 VJCDX01 – Selective Extract for NSDB Databases

*Run Description*

This run is used to extract specials, message, or carrier circuits from NSDB. There are several reasons to use this run. One use of this run is to create a file of circuits for the purpose of synchronization with the TIRKS system. When this is done, the extract from this run is used as input for the synchronization run VJODU03. Another use of this run is to change MCO, CCO, or OCO entries for a given set of circuits. VJCDX01 can be used to extract the candidate circuits so they can be submitted for update on a subsequent run.

This run is processed by scanning either the Circuit Index database (a default), or the In-Effect and Pending databases. The method used is determined by the control card options selected. This run can also be used to extract these same kind of circuits for other reasons. VJCDX01 can be executed as a BMP or DLI run, and has several options to select candidate circuits.

Refer to the run book, 190-534-500, for more detailed information on this run.

*Recommendations*

The run frequency and selection criteria are determined by the user.

*Control Card Information*

The circuits that are to be extracted can be selected by using any of the following circuit attributes as criteria.

**For Special Service Circuits:**

| | |
|---|---|
| CAC | An 8 character code that identifies the Circuit Access Code |
| CLO | A 12 character code that identifies the Circuit Layout Order number |
| CK_SOURCE | A 1 character code that identifies the Circuit Source |
| PREFIX | The first 2 characters of the special service circuit ID |
| SVCODE | A 2 character Service Code that comes from the third and fourth position of the special service circuit ID |
| MOD1 | A 1 character entry that identifies the fifth position of the special service circuit ID |
| MOD2 | A 1 character entry that identifies the sixth position of the special service circuit ID |
| SVCMOD | A 4 character code that identifies the combination of Service Code and Modifier positions |
| AREA | A 3 character code that identifies the area code (NPA) of a telephone numbered special circuit (positions 7, 8, and 9) |
| OFC | A 3 character code that identifies the office unit code of a telephone numbered special circuit (positions 10, 11, and 12) |
| LINE | A 4 character code that identifies the line number code of a telephone numbered special circuit (positions 13, 14, 15, and 16) |
| EXT | A 5 character code that identifies the extension number of a telephone numbered special circuit (positions 17, 18, 19, 20, and 21) |
| SERIAL | Ten characters that come from the 6 character serial number plus a one character delimiter plus a 3 character suffix |
| LOCA6 | The first 6 characters of the location A field |
| LOCA8 | The first 8 characters of the location A field |
| LOCA11 | The entire 11 characters of the location A field |
| LOCZ6 | The first 6 characters of the location Z field |
| LOCZ8 | The first 8 characters of the location Z field |
| LOCZ11 | The entire 11 characters of the location Z field |
| ACTION | A 1 or 2 character code identifying one of the following CLO Actions: A (add), R (rearrange), D (disconnect), RN (rename), C (change) |

| | |
|---|---|
| STATUS | A 1 or 2 character code identifying one of the following CLO Status: P (pending), IE (in-effect), HI |
| MCO | An 11 character code identifying a valid Maintenance Control Office (MCO) |
| OCO | An 11 character code identifying a valid Overall Control Office (OCO) |
| CCO | An 11 character code identifying a valid Circuit Control Office (CCO) |
| RSP | A 2 character code identifying Restoration Priority (RSP) |
| SEG | A 3 character code identifying the segment number for special service circuits (positions 22, 23, and 24 for telephone format; positions 20, 21, and 22 for serial format circuits) |
| LOC6EQ | A 1 character entry to indicate whether the first 6 characters of the location A field are EQUAL TO the first 6 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC8EQ | A 1 character entry to indicate whether the first 8 characters of the location A field are EQUAL TO the first 8 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC11EQ | A 1 character entry to indicate whether the full 11 characters of the location A field are EQUAL TO the full 11 characters of the location Z field (Y = they are equal, N = they are not equal) |
| COMP | A 4 character code that identifies the Assigning Company of a special service circuit (positions 16, 17, 18, and 19) |
| FMT | A 1 character code that identifies the circuit format |
| SSC | An 11 character code that identifies a valid Special Service Center Common Language Location |
| MCN | An 11 character code that identifies the Major Customer Number assigned to a group of special service circuits |
| ACNAC | A 4 character code that identifies the Access Customer Name Abbreviation (ACNA) Code assigned to a special service customer |
| CUSCODE | A 3 character entry that identifies the Customer Code |
| | **For Carrier Circuits:** |
| CAC | An 8 character code that identifies the Circuit Access Code |
| CLO | A 12 character code that identifies the Circuit Layout Order number |
| CK_SOURCE | A 1 character code that identifies the Circuit Source |
| LOCA6 | The first 6 characters of the location A field |
| LOCA8 | The first 8 characters of the location A field |
| LOCA11 | The entire 11 characters of the location A field |

| | |
|---|---|
| LOCZ6 | The first 6 characters of the location Z field |
| LOCZ8 | The first 8 characters of the location Z field |
| LOCZ11 | The entire 11 characters of the location Z field |
| ACTION | A 1 or 2 character code identifying one of the following CLO Actions: A (add), R (rearrange), D (disconnect), RN (rename), C (change) |
| STATUS | A 1 or 2 character code identifying one of the following CLO Status: P (pending), IE (in-effect), HI |
| MCO | An 11 character code identifying a valid Maintenance Control Office (MCO) |
| OCO | An 11 character code identifying a valid Overall Control Office (OCO) |
| CCO | An 11 character code identifying a valid Circuit Control Office (CCO) |
| LOC6EQ | A 1 character entry to indicate whether the first 6 characters of the location A field are EQUAL TO the first 6 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC8EQ | A 1 character entry to indicate whether the first 8 characters of the location A field are EQUAL TO the first 8 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC11EQ | A 1 character entry to indicate whether the full 11 characters of the location A field are EQUAL TO the full 11 characters of the location Z field (Y = they are equal, N = they are not equal) |
| FMT | A 1 character code that identifies the circuit format |
| SSC | An 11 character code that identifies a valid Special Service Center Common Language Location |
| MCN | An 11 character code that identifies the Major Customer Number assigned to a group of special service circuits |
| ACNAC | A 4 character code that identifies the ACNA Code assigned to a special service customer |
| CUSCODE | A 3 character entry that identifies the Customer Code |
| CHANGRP | A 5 character code that identifies the channel group (positions 1 through 5 of the Carrier Circuit ID) |
| CXRTYPE | A 6 character code that identifies the Carrier Type (positions 6 through 11 of the Carrier Circuit ID) |

**For Message Circuits:**

| | |
|---|---|
| TKNO | A 4 character code that identifies the trunk or circuit number of a Message Circuit (first 4 positions of the Message Circuit ID) |
| TGAC | An 8 character code that identifies the Trunk Group Access Code |

| | |
|---|---|
| TFCCLS | A 2 character code that identifies that identifies the Traffic Class of a Message Circuit (positions 5 and 6 of a Message Circuit ID) |
| OFCCLS | A 2 character code that identifies the Office Class of a Message Circuit (positions 7 and 8) |
| TFCUSE | A 2 character code that identifies the Traffic Use of a Message Circuit (positions 9 and 10 of the Message Circuit ID) |
| TFCMOD | A 2 character code found in the 11th and 12th position of the Message Circuit ID |
| PULS | A 2 character code the identifies the Pulsing Code used in a Message Circuit (positions 28 and 29 of the Message Circuit ID) |
| LOCA6 | The first 6 characters of the location A field |
| LOCA8 | The first 8 characters of the location A field |
| LOCA11 | The entire 11 characters of the location A field |
| LOCZ6 | The first 6 characters of the location Z field |
| LOCZ8 | The first 8 characters of the location Z field |
| LOCZ11 | The entire 11 characters of the location Z field |
| ACTION | A 1 or 2 character code identifying one of the following CLO Actions: A (add), R (rearrange), D (disconnect), RN (rename), C (change) |
| STATUS | A 1 or 2 character code identifying one of the following CLO Status: P (pending), IE (in-effect), HI |
| MCO | An 11 character code identifying a valid Maintenance Control Office (MCO) |
| OCO | An 11 character code identifying a valid Overall Control Office (OCO) |
| CCO | An 11 character code identifying a valid Circuit Control Office (CCO) |
| FMT | A 1 character code that identifies the circuit format |
| SSC | An 11 character code that identifies a valid Special Service Center Common Language Location |
| MCN | An 11 character code that identifies the Major Customer Number assigned to a group of special service circuits |
| ACNAC | A 4 character code that identifies the ACNA Code assigned to a special service customer |
| CUSCODE | A 3 character entry that identifies the Customer Code |

In addition to the criteria for selecting candidate circuits, the control card also requires the following user selections

- Execution Mode (XMODE), where input of XMODE=V results in a verification of control card data, or XMODE=N results in normal (no verification) processing of control card data.

- Run Type (RUNTYPE), where input of RUNTYPE=SPECIAL/CARRIER results in the selection of specials and carrier circuits, or RUNTYPE=MESSAGE results in message circuit selection.
- Selection Report option (SEL_RPT), where SEL_RPT=Y results in the production of a report that lists the selected circuits, or SEL_RPT=N results in no Selection Report production. CAUTION: SELECTION OF 'Y' COULD RESULT IN A VERY LARGE REPORT!
- Rejection Report option (REJ_RPT), where REJ_RPT=Y results in the production of a report that lists the rejected circuits, or REJ_RPT=N results in no Rejection Report production. CAUTION: SELECTION OF 'Y' WILL ALWAYS RESULT IN A VERY LARGE REPORT!
- Checkpoint ID, Checkpoint Limit, and Processing Count options are provided for performance optioning.

The goal of the NSDB System Administrator is to provide the ISO group with the control card criteria that uniquely identifies the circuits to be extracted by this run. This is not always any easy task. Care should be taken to use only the minimum number of attributes (called keywords) to uniquely identify the desired circuits. This is important since run time and storage space are directly related to the selection criteria.

The concept of a center name (SSCNAME) is used in the selection criteria for this run. The SSCNAME can be thought of as a header or organizer name for the selected circuits. The SSCNAME can be anything that the user specifies. For example, the SSCNAME could be a CLLI code for a control center, or it could be some other meaningful name to identify the group of circuits.

An equation is used to select the desired circuits. This equation uses the SSCNAME to the left of the = sign, and the circuit attributes that apply to the SSCNAME on the right side of the equation. The circuit attributes are described by:

| | |
|---|---|
| Key Words | Attribute identifiers from the list shown above |
| Value Sets | A user specified name that identifies a grouping of key word values |
| Literal Values | The key word values that are contained in a Value Set |
| Operators | The words AND, OR, IN, and NOTIN. Operators are used to further define the selected circuits by associating key words in various combinations. |

The following example shows how an equation for circuits associated with the SSCNAME of PISCNJ might appear. In addition, the VALUSETs and literal VALUES that were created by the user for this equation are shown.

**The Equation:**

**PISCNJ=LOCA6 OR LOCZ6 IN LOCSET AND ACTION IN ACTSET**

**The Value Sets and their Values:**

*****LOCSET**
**PISCNJ**
**NWBRNJ**

*****ACTSET**
**IE**
**P**

In the equation:

- The SSCNAME is PISCNJ (User created).
- LOCA6, LOCZ6, and ACTION are KEYWORDS (From the list of KEYWORDS).
- OR, IN, AND are OPERATORS (From the list of OPERATORS).

In the Value Sets:

- LOCSET and ACTSET are VALUE SET names (User created).
- PISCNJ and NWBRNJ are literal values that apply in the equation.
- IE and P are literal values (In-Effect and Pending) that apply in the equation.

This equation, with the stated value sets and values, would produce a collection of circuits that:

- Have PISCNJ or NWBRNJ as the first 6 characters of either the A or Z locations in the CLLI code, AND
- Have a status of IE or P

More examples of equations and their use and various options are described in the NSDB run book, BR 190-534-500.

## 8.17 VJCSC08 – TTS Database Load

*Run Description*

The VJCSC08 run is designed to load a TTS database from either a Bellcore–supplied tape (initial delivery or update) or backup created by the extractor program, VJCSM04.

VJCSC08 can be executed as either a DLI or BMP run.

For more information about the VJCSC08 run, refer to BR 190–551–520.

*Recommendations*

Run Frequency – As required.

Prerequisite(s) – VJCSC07 (BR 190–551–519) only during database initialization

Postrequisite(s) – None

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP01.INPUT
Number of steps: 2

## 8.18 VJCSM04 – Database Extract

*Run Description*

The VJCSM04 run is designed to extract data from a TTS database. It will extract specific pieces or the entire database, depending on the control cards supplied to the run. The datasets containing extracted data are compatible with the TTS load run VJCSC08.

VJCSM04 can be executed as either a DLI or BMP run.

For more information about the VJCSM04 run, refer to BR 190–551–521.

*Recommendations*

Run Frequency – As required.

Prerequisite(s) – None

Postrequisite(s) – None

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP01.CARDIN
Number of steps: 4

## 8.19 VJMPM01 – TCM CRON

*Run Description*

This BMP run is used to schedule and trigger IMS transactions based on user input. The transactions are supplied by the control card entries, or from the CRON database where some other subsystem has provided them. Those transactions that are not sensitive to what the calendar date is should be selected via the control cards. These transactions are the ones that run at regular intervals each day, or once every day. An example is the TCM link verification transaction, VJMPMP. (This transaction requires additional IMS security. See the run book BR 190-573-506 for details.)

This BMP run is designed to run continuously each day, with periods of inactivity (called sleep) and execution.

Transactions that are to be scheduled or executed in a IMS control region that is remote to the one that the CRON is running in can also be selected by using the control card.

*Recommendations*

This run is recommended to trigger the TCM link verification process. The link verification should take place several times each hour because it provides the status of the physical links that NSDB has with the external systems. This status is displayed on the VJMPNET format in the CAN-I field.

*Control Card Information*

A maximum of 100 control cards can be used. A maximum of 10 transactions (when provided by the control cards) can be scheduled by the CRON for a given time period.

The control card information shown below is for transactions running in the same IMS control region as the CRON.

| | |
|---|---|
| Transaction Code | Positions 1–8 where the format is A/N |
| Starting Time | Positions 9–12, where the format is HHMM |
| Ending Time | Positions 13–16, where the format is HHMM |
| Interval Time | Positions 17–20, where the format is HHMM |
| Message Text | Positions 21–80, where the format is A/N |

The control card information shown below is for transactions running remotely from the IMS control region of the CRON.

| | |
|---|---|
| Transaction Code | Positions 1–8 where the format is A/N |
| Starting Time | Positions 9–12, where the format is HHMM |
| Ending Time | Positions 13–16, where the format is HHMM |
| Interval Time | Positions 21–80, where the format is A/N |
| Literal MSCNAME= | Positions 21–28, where the value is the Multiple System Coupling (MSC) name |

| | |
|---|---|
| MSC Link Name | Positions 29–36, where the format is A/N |
| Message Text | Positions 37–80, where the format is A/N |

The following example shows the required input to stop this run at a specific time (6:00PM in the example).

VJMPBW 180000000000KILL

| | |
|---|---|
| VJMPBW | The transaction code that schedules the CRON |
| 1800 | The starting time (6PM in military time) |
| 0000 | The ending time |
| 0000 | The interval time |
| KILL | The message text |

Only the Start Time has a value while the Interval Time and End Time are zeros to indicate a one-time-only (per day) transaction.

The following sample shows a TCM link verification and deferred message dequeue process schedule to start at 8AM and end at 6PM. The interval between runs is 2 minutes.

VJMPMP 080018000002

| | |
|---|---|
| VJMPNP | The transaction code that schedules the link verification and dequeue |
| 0800 | The starting time (8AM in military time) |
| 1800 | The ending time (6PM in military time) |
| 0002 | The interval time (every 2 minutes) |

- blank message text

## 8.20 VJMPM03 – Bulk Processing of Messages

*Run Description*

This run is used to resend, or delete messages from the TCM Transaction Log (TLOG) database. The messages to be deleted or resent are selected by the control card input.

*Recommendations*

This BMP type run should be used when necessary to resend or delete large numbers of messages.

*Control Card Information*

This run may be used with multiple control cards. When this is done, the control cards are used in sequence so that processing for the first control card is completed before moving on to the next. The required control card information is shown in the following table.

## VJMPM03 Control Card Options

| COLUMN # | DATA | MEANING |
|---|---|---|
| 1–4 | BRES | The TCM RESEND Command |
| | BDEL | The TCM DELETE Command |
| 5 | blank | Prints information about deleted/resent records |
| | * | Prints information about every TLOG record |
| 6 | A | Selects APPLICATION ERROR |
| | E | Selects TCM ERROR |
| | P | Selects TPAM PARSING ERROR |
| | T | Selects TPAM TRANSLATION ERROR |
| | M | Selects TPAM MAPPING ERROR |
| | G | Selects TPAM GENERAL ERROR |
| | H | Selects HELD MESSAGE |
| | R | Selects RETURNED MESSAGE |
| | K | Selects PENDING ACKNOWLEDGEMENT |
| 8–15 | blank | Messages from all TSYS will be included |
| | TSYS ID | Only messages from this TSYS will be included |
| 17–29 | | Identifies the starting time range. |
| | | Format is YYDDDHHMMSSTH |
| 31–43 | | Identifies the ending time range. |
| | | Format is YYDDDHHMMSSTH |
| 45–49 | | Identifies the checkpoint frequency |
| | | Format is NNNNN |
| 51–55 | | Identifies the maximum # of resent / deleted messages |
| | | Format is NNNNN |
| 57–64 | | Selects the error code to be used if BDEL |
| | | was used in columns 1–4. Otherwise use blank. |

### SAMPLE Control Card input for VJMPM03

BRES H LLSOP 9109712000101 9109812000201 00005 01000

The above sample would produce a run that would:

[1] RESEND messages
(Column 1–4 = BRES, for Bulk Resend)

[2] Print information about only the resent records
(Column 5 = blank)

[3] Select only messages with a HELD status
(Column 6 = H, for Held)

[4] Select only messages from the TSYS called LLSOP
(Column 8–15 = LLSOP, for identifying the TSYS called LLSOP)

[5] Select only messages that were time stamped between 04/07/91 at 12:00.0101 and 04/08/91 at 12:00.0201

(Column 17–29 = 9109712000101, for identifying the start time range, where 91 is the year, 097 is the Julian date for April 7th, and 12000101 is the military time down to hundredths of seconds.)

and

(Column 31–43 = 9109812000201, for identifying the ending time range, where 91 is the year, 098 is the Julian date for April 8th, and 12000201 is the military time down to hundredths of seconds.)

[6] Use a checkpoint frequency of 00005
(Column 45–49 = 00005, for identifying checkpoint frequency)

[7] Apply to a maximum of 1000 messages
(Column 51–55 = 1000, for identifying a maximum of 1000 messages.)

## 8.21 VJMPM04 – Bulk Dequeue of TCM FCIF Messages

*Run Description*

This BMP run is used to dequeue TCM FCIF messages for a user supplied TCM transaction code. The dequeued messages are written to a sequential file. All subject messages must be in FCIF for this run. There is a companion run to enqueue TCM messages called VJMPM05.

*Recommendations*

This run should be used if the IMS buffer is in danger of overflow. These messages can be resubmitted with run VJMPM05.

*Control Card Information*

No control cards are used for this fixdata BMP run.

## 8.22 VJMPM05 – Bulk Enqueue of TCM FCIF Messages

*Run Description*

This BMP run is used to enqueue TCM FCIF messages for a user supplied TCM transaction code. The sequential file from VJMPM04 can be used in this run to place subject FCIF messages on to the IMS queue.

*Recommendations*

This run should be used to place TCM FCIF messages on to the IMS queue. The sequential file created in the VJMPM04 run can be used in this run.

*Control Card Information*

The only selection to be made for this control card is the TCM IMS transaction code name. This transaction code name goes in columns 1–8 and the format is alphanumeric. An example would be VJMPCC1, a transaction used to process Class 1 messages.

## 8.23 VJMPM07 – TCM Statistics Resynchronization

*Run Description*

This run is used to synchronize the statistics in the TCM SEC database. These statistics are the ones that can be viewed under the 'Current Message Status' on the TCM Statistics screen (VJMPSTS). One way the statistics synchronization problem can occur is when TCM paths are deleted and rebuilt while messages exist for those paths.

*Recommendations*

This run should be scheduled frequently since there can be several reasons for synchronization problems. There are no subsequent runs required. This run can be executed in either DLI or BMP, but DLI is highly recommended since any activity on any of the three TCM databases while this job is running can result in errors in the final statistics. It is best not to run this job during periods of heavy system useage.

*Control Card Information*

The purpose of this control card is to override the default checkpoint value. This program gives the user the option of specifying the checkpoint interval by either the number of updates between checkpoints, or the amount of elapsed time between checkpoints. The format used for control card input depends upon the option selected (number of updates or elapsed time). Both formats are shown below:

CHKPFREQ=NNNN (Where NNNN = the # of updates between checkpoint calls)

CHKPTIME=NN (Where NN = the approximate # of minutes between checkpoint calls)

Typically, BMP runs are more concerned with the number of update calls between checkpoints, while DLI runs are more concerned with the elapsed time between checkpoint calls. Because of this, the program uses different default checkpoints that are based on the run execution (BMP or DLI).

The default checkpoints used in this program are:

For BMP – CHKPFREQ=1000 (interval of 1000 updates)

For DLI – CHKPTIME=15 (interval of 15 minutes)

## 8.24 VJODU01 – Bulk Edit for NSDB Databases

*Run Description*

This run is used to identify the DDS, DS1, and CLFI circuits that would fail if sent to ITS/Digital for testing. Reports generated by this run are used to correct and reissue designs that are found in error.

The errors identified by this run can be seen in Appendix A of this guide.

*Recommendations*

This run should be used on a regular basis so that it can be assimilated by the group responsible for correcting the designs. The actual frequency will depend upon the number of errors encountered, and the turn around time for corrections.

*Control Card Information*

The options available for this run are:

[1] Type of edits to be included: (DDS, DS1, CLFI, or a combination)

[2] The partitions and databases the edits are to run against: (Pending, In-Effect, and the partition numbers desired)

[3] Whether or not all partitions are to be included: (All Pending and In-Effect partitions to be included)

[4] Whether or not circuits with an action of D (disconnect) are to be excluded.

SAMPLE CONTROL CARD

```
DISC=N
DDS ALL
DS1 I02
DS1 P03
```

The example above would produce a run that would:

[1] Include disconnected circuits (N = NOT excluded)

[2] Include DDS circuits in ALL partitions of the Pending and In-Effect databases.

[3] Include DS1 circuits in the In-Effect database partition #2

[4] Include DS1 circuits in the Pending database partition #3

## 8.25 VJOQR01 – Batch Query Processing

*Run Description*

This run processes OQS queries that are submitted or saved for deferred execution. These queries can be for one system database, one Manager's Scratch Pad (MSP) database, or multiple MSP databases.

The output reports can be directed to various places depending on the control card options selected. The possible destinations for the reports are shown below:

- Local system printers
- Files into the MSP database
- System databases (subfiles)
- IMS printers (directly or via TSEND)
- Remote destinations via TSEND

*Recommendations*

Run frequency as required. See run book 190-505-015 for more information.

*Control Card Information*

The Control Card information shown here is associated with the STEP01 control card for VJOQR01. Other control cards are also required in subsequent steps, but they are for run performance optioning that the ISO organization is familiar with.

The user supplied information for this run is entered via **keyword paramenter statements.** These statements describe the characteristics of the kind of queries involved, the users of the queries, the databases to be accessed, and the type of output to be created.

Two general types of queries are processed by VJOQR01.

- Submitted Queries

  These are created from OQS using the on-line SUBMIT, BATCH, or BMP commands. These queries are processed by VJOQR01 and the query is then deleted.

- Requested Queries.

  These queries are from the SAVED queries in the OQS databases. These queries are not deleted after VJOQR01 is executed.

**Required Keywords for Submitted Queries**

| | |
|---|---|
| **PRO** | Process Type Identifier. Used to indicate the query is either a Requested (REQ) or Submitted (SUB) type. Use SUB for submitted queries. |
| **DAT** | Database (Name). Used to indicate the database name accessed in the query. Valid database names go here. |
| **QUE** | Query Name. Used to identify the name of the submitted query. The valid query name, or ALL can be entered here. ALL means that all queries for the named database (in DAT) are to be processed. |

**Optional Keywords for Submitted Queries**

**USE** User ID. Used to indicate the ID of the user submitting the query. Multiple users can be specified. The ALL value is the default (i.e., process queries for all users based on the above required keyword values).

**LIS** List option. Used to print an image of the queries being processed. Values are LIS=YES or LIS=NO. Default is NO.

**RUN** Run Mode. Used to identify whether the run is processed in single mode (SIN), or merge mode (MER). If RUN=SIN is selected, all queries are checked for the use of key fields in the query's WHERE statement. If key fields have not been used, the query is not allowed to run in the single (SIN) mode. For queries against MSP databases, RUN=SIN will allow execution of queries for different USER(s)/DATABASE(s). If RUN=MER is used with MSP databases, only those queries for a single (the first named) USER/DATABASE will be executed.

**DEST** IMS destination of output. Used to indicate where the output of VJOQR01 is going. Valid entries include system printers, associated printers, or subfiles.

**OUT** Output Specification. Used to specify what type of query output is desired. Valid entries include extracts to the MSP database (OUT=EXT or OUT=REE), subfile format (OUT=SUB), report format (OUT=REP or OUT=RPT), or a PC subfile format (OUT=PCS). The Report format is the default.

**SUB** Subfile Destinations. This keyword must be used if the Output Specification above is calling for a subfile format (OUT=SUB). Allowed values are single (SIN), or multiple (MUL). See pg 28 of runbook 190-505-015 for more information.

**Required Keywords for Requested Queries**

**PRO** Process Type Identifier. Used to indicate the query is either a Requested (REQ) or Submitted (SUB) type. Use REQ for requested queries.

**DAT** Database (Name). Used to indicate the database name accessed in the query. Valid database names go here.

**QUE** Query Name. Used to identify the name of the submitted query. The valid query name, or ALL can be entered here. ALL means that all queries for the named database (in DAT) are to be processed.

**USE** User ID. Used to indicate the ID of the user submitting the query. Multiple users can be specified. The ALL value is the default (i.e., process queries for all users based on the above required keyword values).

### Optional Keywords for Requested Queries

**LIS** List option. Used to print an image of the queries being processed. Values are LIS=YES or LIS=NO. Default is NO.

**RUN** Run Mode. Used to identify whether the run is processed in single mode (SIN), or merge mode (MER). If RUN=SIN is selected, all queries are checked for the use of key fields in the query's WHERE statement. If key fields have not been used, the query is not allowed to run in the single (SIN) mode. For queries against MSP databases, RUN=SIN will allow execution of queries for different USER(s)/DATABASE(s). If RUN=MER is used with MSP databases, only those queries for a single (the first named) USER/DATABASE will be executed.

**DEST** IMS destination of output. Used to indicate where the output of VJOQR01 is going. Valid entries include system printers, associated printers, or subfiles.

**OUT** Output Specification. Used to specify what type of query output is desired. Valid entries include extracts to the MSP database (OUT=EXT or OUT=REE), subfile format (OUT=SUB), report format (OUT=REP or OUT=RPT), or a PC subfile format (OUT=PCS). The Report format is the default.

**SUB** Subfile Destinations. This keyword must be used if the Output Specification above is calling for a subfile format (OUT=SUB). Allowed values are single (SIN), or multiple (MUL). See pg 28 of runbook 190-505-015 for more information.

## 8.26 VJOQR04 – Batch Query Processing for Partitioned DBS

*Run Description*

This DLI or BMP run is used instead of VJOQR01 if the queries require data from multiple partitions of a given database. If the queries do not require data from multiple partitions of a database, then run VJOQR01 can be used.

Queries can be for one system database, one Manager's Scratch Pad (MSP) database, or multiple MSP databases.

The output reports can be directed to:

- Local system printers
- Files into the MSP database
- System databases (subfiles)
- IMS printers (directly or via TSEND)
- Remote destinations via TSEND

*Recommendations*

Run frequency as required, see 190-505-018 for more information.

*Control Card Information*

(The Control Card information below for VJOQR04 is the same as that for VJOQR01.)

The Control Card information shown here is associated with the STEP01 control card for VJOQR04. Other control cards are also required in subsequent steps, but they are for run performance optioning that the ISO organization is familiar with.

The user supplied information for this run is entered via **keyword paramenter statements.** These statements describe the characteristics of the kind of queries involved, the users of the queries, the databases to be accessed, and the type of output to be created.

Two general types of queries are processed by VJOQR04.

- Submitted Queries

  These are created from OQS using the on-line SUBMIT, BATCH, or BMP commands. These queries are processed by VJOQR04 and the query is then deleted.

- Requested Queries.

  These queries are from the SAVED queries in the OQS databases. These queries are not deleted after VJOQR04 is executed.

### Required Keywords for Submitted Queries

**PRO** Process Type Identifier. Used to indicate the query is either a Requested (REQ) or Submitted (SUB) type. Use SUB for submitted queries.

**DAT** Database (Name). Used to indicate the database name accessed in the query. Valid database names go here.

**QUE** Query Name. Used to identify the name of the submitted query. The valid query name, or ALL can be entered here. ALL means that all queries for the named database (in DAT) are to be processed.

### Optional Keywords for Submitted Queries

**USE** User ID. Used to indicate the ID of the user submitting the query. Multiple users can be specified. The ALL value is the default (i.e., process queries for all users based on the above required keyword values).

**LIS** List option. Used to print an image of the queries being processed. Values are LIS=YES or LIS=NO. Default is NO.

**RUN** Run Mode. Used to identify whether the run is processed in single mode (SIN), or merge mode (MER). If RUN=SIN is selected, all queries are checked for the use of key fields in the query's WHERE statement. If key fields have not been used, the query is not allowed to run in the single (SIN) mode. For queries against MSP databases, RUN=SIN will allow execution of queries for different USER(s)/DATABASE(s). If RUN=MER is used with MSP databases, only those queries for a single (the first named) USER/DATABASE will be executed.

**DEST** IMS destination of output. Used to indicate where the output of VJOQR04 is going. Valid entries include system printers, associated printers, or subfiles.

**OUT** Output Specification. Used to specify what type of query output is desired. Valid entries include extracts to the MSP database (OUT=EXT or OUT=REE), subfile format (OUT=SUB), report format (OUT=REP or OUT=RPT), or a PC subfile format (OUT=PCS). The Report format is the default.

**SUB** Subfile Destinations. This keyword must be used if the Output Specification above is calling for a subfile format (OUT=SUB). Allowed values are single (SIN), or multiple (MUL). See pg 27 of runbook 190-505-018 for more information.

### Required Keywords for Requested Queries

**PRO** Process Type Identifier. Used to indicate the query is either a Requested (REQ) or Submitted (SUB) type. Use REQ for requested queries.

**DAT** Database (Name). Used to indicate the database name accessed in the query. Valid database names go here.

**QUE** Query Name. Used to identify the name of the submitted query. The valid query name, or ALL can be entered here. ALL means that all queries for the named database (in DAT) are to be processed.

**USE** User ID. Used to indicate the ID of the user submitting the query. Multiple users can be specified. The ALL value is the default (i.e., process queries for all users based on the above required keyword values).

**Optional Keywords for Requested Queries**

**LIS** List option. Used to print an image of the queries being processed. Values are LIS=YES or LIS=NO. Default is NO.

**RUN** Run Mode. Used to identify whether the run is processed in single mode (SIN), or merge mode (MER). If RUN=SIN is selected, all queries are checked for the use of key fields in the query's WHERE statement. If key fields have not been used, the query is not allowed to run in the single (SIN) mode. For queries against MSP databases, RUN=SIN will allow execution of queries for different USER(s)/DATABASE(s). If RUN=MER is used with MSP databases, only those queries for a single (the first named) USER/DATABASE will be executed.

**DEST** IMS destination of output. Used to indicate where the output of VJOQR01 is going. Valid entries include system printers, associated printers, or subfiles.

**OUT** Output Specification. Used to specify what type of query output is desired. Valid entries include extracts to the MSP database (OUT=EXT or OUT=REE), subfile format (OUT=SUB), report format (OUT=REP or OUT=RPT), or a PC subfile format (OUT=PCS). The Report format is the default.

**SUB** Subfile Destinations. This keyword must be used if the Output Specification above is calling for a subfile format (OUT=SUB). Allowed values are single (SIN), or multiple (MUL). See pg 27 of runbook 190-505-018 for more information.

## 8.27 VJS1U01 – Update Security Database

*Run Description*

This DLI or BMP run is used to update the NSDB S1 Security database with new GRID definitions and new TTS tables that involve S1 Security. This run should be used whenever a new NSDB release contains changes to the S1 Security GRIDS, or when S1 Security related TTS tables are included in the release.

*Recommendations*

The release package details the usage of this run.

*Control Card Information*

No control cards are required with this run.

## 8.28 VOCDU28 – WFA/C Measurement Database Extract

*Run Description*

VOCDU28 is the WFA/C extract run for the NSDB Measurement database feature. It will read the Closed Trouble Report database (VOTABDD) and will extract/format the trouble reports for loading into the NSDB Measurement database.

VOCDU28 can be executed as either a DLI or BMP run.

For more information about the VOCDU28 run, refer to 190–534–081.

*Recommendations*

Run Frequency – As needed. The run can be used to establish a "base" of closed trouble reports after WFA/C has been converted to Release 1.2. It is also used if the batch interface option is used for the Measurement database feature.

Prerequisite(s) – None

Postrequisite(s) – VJCDU28 (NSDB Measurement Database Load – Maintenance)

Subsequent Run(s) – None

*Control Card Information*

User control cards: STEP01.PGMCNTL and STEP01.IMSCNTL
Number of steps: 1

## 8.29 ZRCDX05 – C1/INV Select and Extract For WFAC/SSC Load

*Run Description*

This batch, DLI, or BMP run is used to extract circuit views from the C1 Inventory module of the TIRKS system. These circuit views include special service circuits, carrier circuits, message circuits, and family circuits. Only one type of circuit can be extracted per execution. This run is used to produce files that are the input to other runs for loading NSDB, WFA/C, and other systems.

*Recommendations*

Run frequency and selection criteria is determined by the user.

*Control Card Information*

The circuits that are to be extracted can be selected by using any of the following circuit attributes as criteria.

**For Special Service Circuits:**

| | |
|---|---|
| CAC | An 8 character code that identifies the Circuit Access Code |
| CLO | A 12 character code that identifies the Circuit Layout Order number |
| PREFIX | The first 2 characters of the special service circuit ID |
| SVCODE | A 2 character Service Code that comes from the third and fourth position of the special service circuit ID |
| MOD1 | A 1 character entry that identifies the fifth position of the special service circuit ID |
| MOD2 | A 1 character entry that identifies the sixth position of the special service circuit ID |
| SVCMOD | A 4 character code that identifies the combination of Service Code and Modifier positions |
| ADMIN | A 2 character code that identifies the Administration Area associated with the record. |
| ALTMCO | An 11 character code that identifies the Alternate Maintenance Control Office (MCO). |
| COMPDATE | A 6 character code that identifies the Completion Date in MMDDYY format. |
| TSP | A 2 character code that identifies the telecommunications Service Priority (TSP) |
| OWNER | A 2 character code that identifies the owner of the data |
| AVN | A 6 character code that identifies the Abbreviated Vendor Name (AVN) |
| CCNA | A 3 character code that identifies the Customer's Carrier Name Abbreviation (CCNA) |

| | |
|---|---|
| ESV | A 12 character code that identifies the Enhanced Service Vendor (ESV) |
| LOCA6 | The first 6 characters of the location A field |
| LOCA8 | The first 8 characters of the location A field |
| LOCA11 | The entire 11 characters of the location A field |
| LOCZ6 | The first 6 characters of the location Z field |
| LOCZ8 | The first 8 characters of the location Z field |
| LOCZ11 | The entire 11 characters of the location Z field |
| ACTION | A 1 or 2 character code identifying one of the following CLO Actions: A (add), R (rearrange), D (disconnect), RN (rename), C (change), or the CAC status: IE (in-effect) |
| MCO | An 11 character code identifying a valid Maintenance Control Office (MCO) The use of this attribute slows the run considerably. |
| OCO | An 11 character code identifying a valid Overall Control Office (OCO) |
| CCO | An 11 character code identifying a valid Circuit Control Office (CCO) |
| RSP | A 2 character code identifying Restoration Priority (RSP) |
| SEG | A 3 character code identifying the segment number for special service circuits (positions 22,23, and 24 for telephone format; positions 20, 21, and 22 for serial format circuits) |
| LOC6EQ | A 1 character entry to indicate whether the first 6 characters of the location A field are EQUAL TO the first 6 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC8EQ | A 1 character entry to indicate whether the first 8 characters of the location A field are EQUAL TO the first 8 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC11EQ | A 1 character entry to indicate whether the full 11 characters of the location A field are EQUAL TO the full 11 characters of the location Z field (Y = they are equal, N = they are not equal) |
| COMP | A 4 character code that identifies the Assigning Company of a special service circuit (positions 16, 17, 18, and 19) |
| FMT | A 1 character code that identifies the circuit format |
| MCN | An 11 character code that identifies the Major Customer Number assigned to a group of special service circuits |
| ACNAC | A 4 character code that identifies the Access Customer Name Abbreviation (ACNA) Code assigned to a special service customer |

**For Carrier Circuits:**

| | |
|---|---|
| CAC | An 8 character code that identifies the Circuit Access Code |
| CLO | A 12 character code that identifies the Circuit Layout Order number |
| PREFIX | The first 2 characters of the special service circuit ID |
| SVCODE | A 2 character Service Code that comes from the third and fourth position of the special service circuit ID |
| MOD1 | A 1 character entry that identifies the fifth position of the special service circuit ID |
| MOD2 | A 1 character entry that identifies the sixth position of the special service circuit ID |
| SVCMOD | A 4 character code that identifies the combination of Service Code and Modifier positions |
| ADMIN | A 2 character code that identifies the Administration Area associated with the record. |
| ALTMCO | An 11 character code that identifies the Alternate Maintenance Control Office (MCO). |
| COMPDATE | A 6 character code that identifies the Completion Date in MMDDYY format. |
| TSP | A 2 character code that identifies the telecommunications Service Priority (TSP) |
| OWNER | A 2 character code that identifies the owner of the data |
| AVN | A 6 character code that identifies the Abbreviated Vendor Name (AVN) |
| CCNA | A 3 character code that identifies the Customer's Carrier Name Abbreviation (CCNA) |
| ESV | A 12 character code that identifies the Enhanced Service Vendor (ESV) |
| LOCA6 | The first 6 characters of the location A field |
| LOCA8 | The first 8 characters of the location A field |
| LOCA11 | The entire 11 characters of the location A field |
| LOCZ6 | The first 6 characters of the location Z field |
| LOCZ8 | The first 8 characters of the location Z field |
| LOCZ11 | The entire 11 characters of the location Z field |
| ACTION | A 1 or 2 character code identifying one of the following CLO Actions: A (add), R (rearrange), D (disconnect), RN (rename), C (change), or the CAC status: IE (in-effect) |
| MCO | An 11 character code identifying a valid Maintenance Control Office (MCO) The use of this attribute slows the run considerably. |

| | |
|---|---|
| OCO | An 11 character code identifying a valid Overall Control Office (OCO) |
| CCO | An 11 character code identifying a valid Circuit Control Office (CCO) |
| RSP | A 2 character code identifying Restoration Priority (RSP) |
| SEG | A 3 character code identifying the segment number for special service circuits (positions 22, 23, and 24 for telephone format; positions 20, 21, and 22 for serial format circuits) |
| LOC6EQ | A 1 character entry to indicate whether the first 6 characters of the location A field are EQUAL TO the first 6 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC8EQ | A 1 character entry to indicate whether the first 8 characters of the location A field are EQUAL TO the first 8 characters of the location Z field (Y = they are equal, N = they are not equal) |
| LOC11EQ | A 1 character entry to indicate whether the full 11 characters of the location A field are EQUAL TO the full 11 characters of the location Z field (Y = they are equal, N = they are not equal) |
| FMT | A 1 character code that identifies the circuit format |
| MCN | An 11 character code that identifies the Major Customer Number assigned to a group of special service circuits |
| ACNAC | A 4 character code that identifies the Access Customer Name Abbreviation (ACNA) Code assigned to a special service customer |
| CHANGRP | A 5 character code that identifies the channel group (positions 1 through 5 of the Carrier Circuit ID) |
| CXRTYPE | A 6 character code that identifies the Carrier Type (positions 6 through 11 of the Carrier Circuit ID) |
| | **For Message Circuits:** |
| TKNO | A 4 character code that identifies the trunk or circuit number of a Message Circuit (first 4 positions of the Message Circuit ID) |
| TGAC | An 8 character code that identifies the Trunk Group Access Code |
| TFCCLS | A 2 character code that identifies that identifies the Traffic Class of a Message Circuit (positions 5 and 6 of a Message Circuit ID) |
| OFCCLS | A 2 character code that identifies the Office Class of a Message Circuit (positions 7 and 8) |
| TFCUSE | A 2 character code that identifies the Traffic Use of a Message Circuit (positions 9 and 10 of the Message Circuit ID) |
| TFCMOD | A 2 character code found in the 11th and 12th position of the Message Circuit ID |
| PULS | A 2 character code the identifies the Pulsing Code used in a Message Circuit (positions 28 and 29 of the Message Circuit ID) |

| | |
|---|---|
| LOCA6 | The first 6 characters of the location A field |
| LOCA8 | The first 8 characters of the location A field |
| LOCA11 | The entire 11 characters of the location A field |
| LOCZ6 | The first 6 characters of the location Z field |
| LOCZ8 | The first 8 characters of the location Z field |
| LOCZ11 | The entire 11 characters of the location Z field |
| ACTION | A 1 or 2 character code identifying one of the following CLO Actions: A (add), R (rearrange), D (disconnect), RN (rename), C (change) |
| STATUS | A 1 or 2 character code identifying one of the following CLO Status: P (pending), IE (in-effect), HI |
| MCO | An 11 character code identifying a valid Maintenance Control Office (MCO) The use of this attribute slows the run considerably. |
| OCO | An 11 character code identifying a valid Overall Control Office (OCO) |
| CCO | An 11 character code identifying a valid Circuit Control Office (CCO) |
| FMT | A 1 character code that identifies the circuit format |
| SSC | An 11 character code that identifies a valid Special Service Center Common Language Location |
| MCN | An 11 character code that identifies the Major Customer Number assigned to a group of special service circuits |
| ACNAC | A 4 character code that identifies the ACNA Code assigned to a special service customer |
| CUSCODE | A 3 character entry that identifies the Customer Code |
| CAC | An 8 character code that identifies the Circuit Access Code |
| CLO | A 12 character code that identifies the Circuit Layout Order number |
| COMPDATE | A 6 character code that identifies the Completion Date in MMDDYY format. |
| TSP | A 2 character code that identifies the telecommunications Service Priority (TSP) |
| OWNER | A 2 character code that identifies the owner of the data |
| AVN | A 6 character code that identifies the Abbreviated Vendor Name (AVN) |
| CCNA | A 3 character code that identifies the Customer's Carrier Name Abbreviation (CCNA) |
| ESV | A 12 character code that identifies the Enhanced Service Vendor (ESV) |

In addition to the criteria for selecting candidate circuits, the control card also requires the following user selections

- Execution Mode (XMODE), where input of XMODE=V results in a verification of control card data, or XMODE=N results in normal (no verification) processing of control card data.
- Run Type (RUNTYPE), where input of RUNTYPE=S selects specials, RUNTYPE=C selects carrier circuits, or RUNTYPE=M selects message circuits.
- Selection Report option (SEL_RPT), where SEL_RPT=Y results in the production of a report that lists the selected circuits, or SEL_RPT=N results in no Selection Report production. CAUTION; SELECTING 'Y' COULD RESULT IN A VERY LARGE REPORT!
- Rejection Report option (REJ_RPT), where REJ_RPT=Y results in the production of a report that lists the rejected circuits, or REJ_RPT=N results in no Rejection Report production. CAUTION; SELECTING 'Y' WILL ALWAYS RESULT IN A VERY LARGE REPORT!
- Checkpoint ID, Checkpoint Limit, Processing Count, and Debug selections are required.

To select the candidate circuits and processing options for this run, make control card entries for these steps:

| | |
|---|---|
| STEP01.PGMCNTL DD | Make entries for CHKPTID, CHKPTLIMIT, and PROCESSCNT |
| STEP01.SYSIN DD | Make entries for XMODE, RUNTYPE, SEL_RPT, REJ_RPT, and DEBUG |
| STEP01.EXTRACT DD | Make entries for SSCNAME (see explanation below) |
| STEP01.VALUSET | Make entries for desired valuesets (see explanation below) |

The goal of the NSDB System Administrator is to provide the ISO group with the control card criteria that uniquely identifies the circuits to be extracted by this run. This is not always any easy task. Care should be taken to use only the minimum number of attributes (called keywords) to uniquely identify the desired circuits. This is important since run time and storage space are directly related to the selection criteria.

The concept of a center name (SSCNAME) is used in the selection criteria for this run. The SSCNAME can be thought of as a header or organizer name for the selected circuits. The SSCNAME can be anything that the user specifies. For example, the SSCNAME could be a CLLI code for a control center, or it could be some other meaningful name to identify the group of circuits.

An equation is used to select the desired circuits. This equation uses the SSCNAME to the left of the = sign, and the circuit attributes that apply to the SSCNAME on the right side of the equation. The circuit attributes are described by:

| | |
|---|---|
| Key Words | Attribute identifiers from the list shown above |
| Value Sets | A user specified name that identifies a grouping of key word values |
| Literal Values | The key word values that are contained in a Value Set |
| Operators | The words AND, OR, IN, and NOTIN. Operators are used to further define the selected circuits by associating key words in various combinations. |

The following example shows how an equation for circuits associated with the SSCNAME of PISCNJ might appear. In addition, the VALUSETs and literal VALUES that were created by the user for this equation are shown.

**The Equation:**

**PISCNJ=LOCA6 OR LOCZ6 IN LOCSET AND ACTION IN ACTSET**

**The Value Sets and their Values:**

*****LOCSET**
**PISCNJ**
**NWBRNJ**

*****ACTSET**
**IE**
**P**

In the equation:

- The SSCNAME is PISCNJ (User created).
- LOCA6, LOCZ6, and ACTION are KEYWORDS (From the list of KEYWORDS).
- OR, IN, AND are OPERATORS (From the list of OPERATORS).

In the Value Sets:

- LOCSET and ACTSET are VALUE SET names (User created).
- PISCNJ and NWBRNJ are literal values that apply in the equation.
- IE and P are literal values (In-Effect and Pending) that apply in the equation.

This equation, with the stated value sets and values, would produce a collection of circuits that:

- Have PISCNJ or NWBRNJ as the first 6 characters of either the A or Z locations in the CLLI code, AND
- Have a status of IE or P

More examples of equations and their use and various options are described in the TIRKS run book, BR 190-535-507.

## 8.30 ZRCDX07 – Extract Selected Circuits For OPS DB Load

*Run Description*

This BMP or DLI run is used with the output of TIRKS run ZRCDX05 to extract orders from the TIRKS system. The prerequisite run provides the list of CAC/CLO pairs that is used to extract the circuits.

*Recommendations*

Can be used as needed. This run is also part of the installation process of NSDB databases. Refer to BSP 756–586–001 for the TIRKS EDIIS table with the EVENT NAME of BDIST, and to the TAS notes required for the NMA load.

*Control Card Information*

Control Card choices for this run are limited to internal buffer size options, output file compression option, and EDIIS distribution or index page selections.

# Other Tools for NSDB System Administration
## CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# 9. Other Tools for System Administration

This chapter, "Other Tools for System Administration", contains an assortment of information that will be useful to the NSDB system administrator. The information provided in this chapter is as follows:

- "A Functional Guide to Documentation for NSDB System Administration" provides descriptions of the Bellcore documents that support the tasks involved with NSDB system administration.
- The "Cross–Reference to TIRKS Core Documents" correlates outdated NSDB document numbers to the current TIRKS document numbers for a number of TIRKS and TCM documents.
- "The TIRKS System Tables Required by NSDB" gives the NSDB system administrator an appreciation for the Bellcore TIRKS® system tables that support one of the primary sources of provisioning data for NSDB.
- "The SOAC Tables Required by NSDB" provides an overview of the Service Order Analysis and Control (SOAC) system screens and tables that support the other major source of provisioning data for NSDB.
- "TCM Online Formats" is a brief section that introduces the system administrator to the TCM screens which will be heavily used in his/her work.
- An overview of the "Open Query System (OQS)" is included since OQS queries are used by the NSDB system administrator to view the NSDB and TCM databases.
- A section is provided to explain "How NSDB Derives a Line Record from the TIRKS Circuit Detail".
- Another section in this chapter explains the "Procedure for Turning Up the MPI Feature".
- "Archive and Retrieval Capabilities for NSDB" is a detailed account of the functionality that permits the removal of data from online databases for storage (archive) on magnetic or cassette tape with the option to review this data at a later time. It is a recent enhancement to the NSDB software.

TIRKS is a registered trademark of Bellcore.

## 9.1 A Functional Guide to Documentation for NSDB System Administration

The following tables list documentation related to NSDB system administration functions.

**Table 9–1.** Functional Guide to Documentation
(1 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| Setting up and Maintaining the NSDB Interfaces | NSDB System Administrator's Guide 190-534-005 | The NSDB System Administrator's Guide describes online screens and tables for setup and maintenance activities. It provides overviews of the NSDB TTS tables, descriptions of all fields appearing on these tables, and lists table attributes. The tools available to the NSDB System Administrator are discussed as well. |
| | Operations Systems Interface Administration Guide 190-518-001 | This book describes setup and trouble shooting procedures for operations systems interfaces. The cross-product interfaces include WFA/C, WFA/DO, WFA/DI, and NSDB. |
| | NSDB Product List | The NSDB Product List gives a listing of all the Bellcore documentation associated with NSDB. Runbooks are listed by BR# and run title. (The NSDB Product List should be available through your documentation coordinator.) |
| | TIRKS Table System (TTS) Position Guide 756–551–700 | This document explains the TTS core subsystem formats. It is valuable as a reference for experienced administrators, and is recommended reading for new administrators. |
| | NSDB Database Administrator's Guide 190-534-001 | This document provides reference information about the NSDB databases. It is targeted at the ISO personnel responsible for installing and maintaining NSDB databases. |

**Table 9–1.** Functional Guide to Documentation
(2 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| Setting up and Maintaining the NSDB Interfaces (continued) | Operations Cross-Product Transaction Guide 190-500-001 | This document provides reference transaction information associated with work flows across multiple products. This book includes transaction definitions, TCM paths, interproduct formats, and interproduct TTS tables. |
| | NSDB Installation Guide 190-534-004 | This document provides instructions and reference information about the installation of NSDB releases. This book is primarily intended for companies that are installing NSDB for the first time. |

**Table 9–1.** Functional Guide to Documentation
(3 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| Intersystem Communications Administration | NSDB System Administrator's Guide 190-534-005 | The NSDB System Administrator's Guide describes the tables and screens for both setup and maintenance activities. It also describes the tools available to the NSDB System Administrator. The TCM Network Administration screens are included, and populated screen examples are provided as well. |
| | TIRKS Communication Module (TCM) Overview 756–573–301 | This document gives a "big picture" of the TCM system, and explains the purpose of each of its components. |
| | TIRKS Communication Module (TCM) Route Administration (RA) User Manual 756–573–302 | This document describes the "heart of TCM", the Routing Administration (RA) component. Detailed discussions are provided for Input Header, Class 1 and Class 3 Message, Acknowledgement, Error, and Dequeue processing. |
| | TIRKS Communication Module (TCM) Message Administration (MA) User Manual 756–573–303 | This document discusses the tools used to monitor and correct messages routed through TCM. The message (VJMPMSG) and test (VJMPTST) formats are described here. |
| | TIRKS Communication Module (TCM) Network Administration (NA) User Manual 756–573–304 | This document describes the tools used to monitor and maintain the network of communication channels that connect to TCM. The network administration (VJMPNET) and network statistics (VJMPSTS) formats are described here. |

**Table 9–1.** Functional Guide to Documentation
(4 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| Intersystem Communications Administration (continued) | Operations Cross-Product Transaction Guide 190-500-001 | This document provides reference transaction information associated with work flows across multiple products. This book includes transaction definitions, TCM paths, interproduct formats, and interproduct TTS tables. |
| | TIRKS Communication Module (TCM) Translation Administration (TA) User Manual 756–573–305 | This document describes the tools available to translate data that is incoming to, or outgoing from TCM. The use of TCM translation rules is discussed as well. |
| | TIRKS Communication Module (TCM) CRON User Manual 756–573–382 | This document describes the TCM CRON component. TCM CRON is used to schedule IMS transactions for TCM and other applications within the NSDB system. |
| | NSDB Contract Dictionary 190-534-321 | This document describes data that is sent to and from NSDB from various provisioning and operations systems. It also describes the construction of the messages, and gives definitions of the individual data tags that are passed over the interfaces. |
| | Flexible Computer Interface Format (FCIF) Description and Syntax 756–551–022 | This document describes FCIF in both technical and general terms. |

**Table 9–1.** Functional Guide to Documentation
(5 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| Maintaining NSDB Data Integrity | NSDB System Administrator's Guide 190-534-005 | This document provides information about the tools used to correct database errors (see VJDBADM). An overview of the table settings required by source systems (TIRKS and SOAC) is also provided. |
| | NSDB Contract Dictionary 190-534-321 | This document describes the data that is sent to and from NSDB from various provisioning and operations systems. It also describes the construction of the messages, and gives definitions of the individual data tags that are passed over the interfaces. |
| | NSDB Database Modifier User Manual 190-534-307 | This document describes how to use the NSDB Database Modifier format (VJDIDBM) to correct, update, add, or delete data for the NSDB databases. Please note that this document is restricted, and is available only through Bellcore's TSC. The use of VJDIDBM should be limited to individuals who fully understand the format and its potential for damage when used inappropriately. |
| | C-1 Location Code Information Format (VJCRLOC) 756–554–740 | This document describes how to use the VJCRLOC format to maintain the Location Reference database (LOCREF). This activity is critical to ensure synchronization between TIRKS and the NSDB LOCREF databases. |

**Table 9–1.** Functional Guide to Documentation
(6 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| NSDB Security Administration | NSDB Security Administrator's Guide 190-534-003 | This document describes how NSDB's S1 security relates to IMS/VS system security. This book also describes how the individual subsystems and formats within NSDB are secured. It is restricted and is only available through Bellcore's TSC. |

**Table 9–1.** Functional Guide to Documentation
(7 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| NSDB Error Processing | NSDB System Administrator's Guide 190-534-005 | This document describes the categories of errors and associated error directories. It also describes the routing of error messages using the ODB EXCEPTIONS, ODB MCO OPTIONS, and the ODB CO OPTIONS tables. |
| | NSDB Format to Transaction Cross Reference Guide 190-534-002 | This book presents information about each transaction in the current SYSGEN. The information includes the subsystem associated with the transaction, PSB name, format name(s), commands, PF keys, average DL/I calls per transaction, and a description of the transaction. Message switches and MSC links are identified in the transaction's description field.<br><br>This document is restricted, and is available only through Bellcore's TSC. |
| | TIRKS Communication Module (TCM) Message Administration (MA) User Manual 756–573–303 | This document describes how TCM message errors can be handled. The TCM message format (VJMPMSG) is discussed as well. |
| | Operations Cross-Product Transaction Guide 190-500-001 | This document provides reference transaction information associated with work flows across multiple products. This book includes transaction definitions, TCM paths, interproduct formats, and interproduct TTS tables. |

**Table 9–1.** Functional Guide to Documentation
(8 of 8)

| ADMINISTRATIVE FUNCTION | BELLCORE DOCUMENTS | DESCRIPTION |
|---|---|---|
| NSDB Error Processing (continued) | NSDB Online Message Directory 190-534-400 | This book describes application and system errors that can appear when using NSDB online transactions. |
| | TIRKS Table System (TTS) Online Message Directory 756–551–262 | This book describes application and system errors that can appear when using TTS online transactions. |
| | TIRKS Communication Module (TCM) Online Message Directory 756–573–260 | This book describes application and system errors that can appear when using TCM online transactions. |
| | Circuit Inventory Reference Data Module (C-1 REF) Online Message Directory 756–580–260 | This book describes application and system errors that can appear when using C-1 REF online transactions. |
| | NSDB Exception Notice Directory 190-534-404 | This book describes the exception notices that can appear as a result of NSDB application errors. |

## Table 9–2. Cross-Reference to TIRKS Core Documents

| Document Name | Former NSDB Document Number: BR 190-534- | TIRKS Document Number: BR 756- | Most Recent Issue of TIRKS Document | TIRKS Release |
|---|---|---|---|---|
| *TTS Position Guide* | 100 | 551-700 | Issue 6 (11/91) | 16.0 |
| *TCM Overview* | 301 | 573-301 | Issue 13 (11/92) | 16.1 |
| *TCM Routing Administration (RA) User Manual* | 302 | 573-302 | Issue 13 (11/92) | 16.1 |
| *TCM Message Administration (MA) User Manual* | 303 | 573-303 | Issue 12 (11/92) | 16.1 |
| *TCM Network Administration (NA) User Manual* | 304 | 573-304 | Issue 16 (11/92) | 16.1 |
| *TCM Translation Administration (TA) User Manual* | 305 | 573-305 | Issue 10 (11/92) | 16.1 |
| *TCM Cron User Manual* | 306 | 573-382 | Issue 3 (11/91) | 16.0 |
| *FCIF Description and Syntax* | 308 | 551-022 | Issue 3 (11/92) | 16.1 |
| *C-1 Location Code Information Format* | 315 | 554-740 | Issue 15 (11/92) | 16.1 |
| *TIRKS Table System (TTS) On-Line Message Directory* | 401 | 551-262 | Issue 3 (11/86) | 14.3 |
| *TIRKS Communication Module (TCM) On-Line Message Directory* | 402 | 573-260 | Issue 14 (11/92) | 16.1 |
| *Circuit Inventory Reference Data Module (C1/REF) On-Line Message Directory* | 403 | 580-260 | Issue 1 (9/84) | 14.0 |

**NOTE –** As of NSDB Release 3.2, information that was formerly contained in the *NSDB Reference Data TTS Encyclopedia* (BR 190–534–006) and the *NSDB Reference Data TTS Field Dictionary* (BR 190–534–007) has been integrated into the *NSDB System Administrator's Guide* (BR 190–534–005).

## 9.2 The TIRKS System Tables Required By NSDB

Four of the tables that support the TIRKS system must be populated to enable NSDB to receive data from the Event Driven Interproduct Interface System (EDIIS). The NSDB System Administrator must ensure that these entries exist. The overall responsibility for the TIRKS system tables may vary from region to region. It is possible that the NSDB System Administrator will need to coordinate the input and verification of the NSDB-related entries with the EDIIS or TIRKS System Administrator(s).

The four TIRKS system tables are:

- EDIIS EVENT FUNC
- EDIIS NSDB OPTS
- EDIIS OPTION CD
- EDIIS UCT

The EDIIS EVENT FUNC table identifies the 'events' that serve as triggers for the sending of data to NSDB and other systems. Three events can be identified for NSDB purposes. The first event, ISSUWD, is identified so that NSDB can receive data from the TIRKS system whenever a WORD document is issued. Data received as a result of the ISSUWD event is used by ITS and the NMA system. The second event, NMADATA, is identified so that NSDB can receive data specifically for the NMA system. BCCs that do not use the NMA system do not need to identify the event NMADATA on the EDIIS EVENT FUNC table. The third event, ISSUIX, can be identified so that the index page from the TIRKS system is sent to NSDB. This feature is optional, and if ISSUIX is not identified on the EDIIS EVENT FUNC table, NSDB will continue to construct multipoint circuit views as in previous releases.

The EDIIS NSDB OPTIONS table is used to identify to EDIIS the classes of message circuits to be sent to NSDB.

The EDIIS OPTION CD table is used by EDIIS to determine which administrative areas will use the interface to NSDB. The three possible administrative areas are Message, Carrier, and Specials. The BCC may select any or all of these three types by making the appropriate entries on the EDIIS OPTION CD table.

The EDIIS UCT table identifies the Logical Terminal (LTERM) used for exception notices and the buffer sizes used for the input/output and TPAM buffers. While this table requires no entries that are unique for NSDB, the EDIIS UCT table must be populated with the appropriate entries to ensure the continued flow of data from the TIRKS system.

### 9.2.1 EDIIS EVENT FUNC

**INTERFACE ID = TIRKS-NSDB-ITS/Digital**
**TIRKS-NSDB-NMA**
**TIRKS-NSDB (For MPI Index Page data)**

This table defines the events (ISSUWD, NMADATA, and ISSUIX) to EDIIS that trigger data to be sent to TCM for NSDB. The events that are required and identified on this table depend upon the BCC's usage of ITS and the NMA system, and whether or not the TIRKS system Multipoint Index data is needed.

For those BCCs that have an interface between the TIRKS system, NSDB, and ITS, the event ISSUWD must be defined in the EDIIS EVENT FUNC table. If the TIRKS system, NSDB, and the NMA system interface is used, then the event NMADATA must be defined in the EDIIS EVENT FUNC table. If TIRKS system Multipoint Index data is to be used to create an index page, then the event ISSUIX must be defined in the EDIIS EVENT FUNC table.

The NSDB System Administrator should coordinate the creation or verification of the EDIIS EVENT FUNC table(s) with the EDIIS Administrator. These tables are preloaded and set up by the EDIIS Administrator (refer to the *EDIIS Installation Guide*, BR 756-586-022, for additional information). The NSDB System Administrator is only responsible only for the MFD/SEC field*. This field is user–defined and should be entered or verified at this time.

* The MFD/SEC field is an 8 character field for identifying the Message Format Descriptor (MFD) to be used for LOG functions, or the TIRKS Communications Module (TCM) System Entity Code (SEC) to be used for certain XMIT functions.

```
COMMAND          ***  TIRKS-TTS DATA SCREEN  ***                  /FOR


TABLE NAME:  EDIIS EVENT FUNC  TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS:
NOTE: EDIIS EVENT FUNCTIONS                      REL/LEV:        MOD:
   FIELD     FIELD
   NAME      VALUE
   SEQ#          *
   FUNCTION          *
   SUBSYSTM          *
   MFD/SEC           *
   MODULE            *









TTS200E FIND UNSUCCESSFUL DATA NOT FOUND DESCRIPTORS MAPPED
```

**Figure 9–1.** EDIIS EVENT FUNC Table

VERIFICATION OF THE EDIIS EVENT FUNC

The following steps should be taken to verify the EDIIS EVENT FUNC table for each event required for NSDB:

1. Display the EDIIS Event Table Administration Screen by typing /FOR ZRIIETA and pressing the ENTER key.
2. On the returned screen, type the desired event (ISSUWD, NMADATA, or ISSUIX) in the EVENT NAME field. Perform a FIND by pressing the PF1 function key.
3. On the returned screen, you should find 'NSDB' under the SUBSYS field. Move your curser to the 'N' in NSDB and press the ENTER key.
4. Verify that the value in the MFD/SEC field is correct for NSDB. The MFD/SEC value should be an 8–character alphanumeric for the System Entity Code (SEC) that identifies NSDB. If the value is missing or incorrect, you can change it by typing a 'C' to the left of the MFD/SEC field, typing in the correct value, and pressing the PF5 function key to UPDATE this table.

### 9.2.2 EDIIS NSDB OPTS

**INTERFACE ID = TIRKS-NSDB-ITS/Digital**

This table is required in order to receive message trunks from the TIRKS system, and it acts as a validation table for message circuits. It is used to identify to EDIIS the classes of message circuits to be sent to NSDB. This is done via the TUC field (Traffic Use Code) and the PULSING field. All TUC codes must be reflected in this table. The TUC field is 2 characters long and is taken from the 5th and 6th characters of the Trunk Group Descriptor portion of the message circuit ID. The PULSING field is also 2 characters long and is taken from the 29th and 30th characters of the CLCI codes for message circuits.

```
COMMAND          ***  TIRKS-TTS DATA SCREEN  ***                    /FOR


TABLE NAME:  EDIIS NSDB OPTS   TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                # OF RECORDS: 000
NOTE: VALIDATION TABLE FOR MSG CKTS FOR NSDB   REL/LEV:       MOD: Y
   FIELD     FIELD
   NAME      VALUE
   TUC         *
   PULSING     *









TTS200E FIND UNSUCCESSFUL DATA NOT FOUND DESCRIPTORS MAPPED
```

**Figure 9–2.** EDIIS NSDB OPTS Table

## EDIIS NSDB OPTS TABLE DESCRIPTIONS

TUC

Full Name: Traffic Use Code

Definition: This code identifies the find of traffic (interexchange, common channel signaling, etc.) that will be carried over a circuit. The TUC value represents the 5th and 6th characters of the Trunk Group Descriptor portion of the message circuit ID.

Data Content: 2 A/N

Valid Entries: Valid Traffic Use Codes

Examples: TO, AD

PULSING

Full Name: Pulsing Code

Definition: This code indicates the type and direction of pulsing signals (other than supervisory signals) between location A and location Z. The type and direction of the pulse code is identified by positions 29 and 30 of the CLCI codes for message trunks or special service trunks. The 29th character represents the type of pulse outgoing from location A to location Z. The 30th character represents the type of pulsing in the Z to A direction.

Data Content: 2 A/N (Optional)

– 1st character = pulsing code in A to Z direction

– 2nd character = pulsing code in Z to A direction

Valid Entries: A, C, D, F, J, K, M, P, R, S, V, 5, 6, –, or blank

## CREATION OF THE EDIIS NSDB OPTS TABLE

To complete this table, follow the steps below:

1. Type /FOR ZRTDSO and press the ENTER key.
2. On the returned screen, type EDIIS NSDB OPTS in the TABLE NAME field, and press the PF1 function key.
3. Type in the desired TUC and PULSING entries, and press the PF4 function key to ADD the record.
4. This screen also supports the SCAN option. To use this option, type SCAN in the COMMAND field and press the PF1 function key.

### 9.2.3 EDIIS OPTION CD

**INTERFACE ID = TIRKS-NSDB-ITS/Digital**
**= TIRKS-NSDB-NMA/Facilities**
**= TIRKS/NSDB-NMA/Switch**

This table identifies the BCC Administrative Areas associated with the target data that EDIIS will route to the NMA system via NSDB. The BCC Administrative Area is represented by the first two characters of the Circuit Layout Order (CLO). EDIIS also uses this table to determine if the EDIIS/NMA interface is active. In addition, EDIIS checks the Assignment Responsibility of this data against another EDIIS table called F1DATA ASGTRESP. The Assignment Responsibility table specifies the organization within the BCC that is responsible for selecting inventoried equipment and facilities, and the issuance of orders. Lists of all valid BCC Administrative Areas and Assignment Responsibility Areas are available to the NSDB System Administrator. The SCANT command performed on the TIRKS TTS table OPTION FLAGS will provide a list of the BCC Administrative Areas. The SCAN command performed on the TIRKS TTS table F1DATA ASGTRESP will provide a list of the Assignment Responsibility Areas.

The NSDB System Administrator must complete two versions of the EDIIS OPTION CD table in order to activate the TIRKS-NSDB-NMA interface, and the TIRKS-NSDB-ITS/Digital interface. Without these tables, no data will be passed to the NMA system and ITS respectively. The EDIIS table F1DATA ASGTRESP is maintained by the EDIIS Administrator.

```
COMMAND          ***  TIRKS-TTS DATA SCREEN  ***                  /FOR


TABLE NAME:  EDIIS OPTION CD   TABLE KEY:                    ADMIN AREA:
TABLE RECORD KEY:                                 # OF RECORDS: 0000
NOTE: VALIDATES THE OPTION CODES                 REL/LEV:           MOD: Y
   FIELD     FIELD
   NAME      VALUE
   OPT CODE   *









TTS200E FIND UNSUCCESSFUL DATA NOT FOUND DESCRIPTORS MAPPED
```

**Figure 9–3.** EDIIS OPTION CD Table

## EDIIS OPTION CD TABLE FIELD DESCRIPTIONS

OPT

Full Name: Option Code

Definition: Identifies the Administrative Area or circuit type to be passed to NSDB.

Data Content: 1 A

Valid Entries: M, C, or S

M = Message trunks

C = Carrier circuits

S = Special services circuits

## CREATION OF THE EDIIS OPTION CD TABLE

Two versions of this table are required if both the TIRKS-NSDB-ITS/Digital and the TIRKS-NSDB-NMA/Switch interfaces are in use. One version has a TABLE KEY of 'NSDB' with a TABLE RECORD KEY of 'M'. The other version has a TABLE KEY of 'NMA AA' with a TABLE RECORD KEY of 'ADMIN AREA'. The TIRKS-NSDB-ITS/Digital interface has a TABLE KEY of 'NSDB' and TABLE RECORD KEYs of 'S'and 'C'.

Follow the steps below to complete this table:

1. Type /FOR ZRTDSO and press the ENTER key.
2. On the returned screen, type EDIIS OPTION CD in the TABLE NAME field and NSDB in the TABLE KEY field. Perform a FIND by pressing the PF1 function key.
3. The returned screen is used to identify the types of circuits that EDIIS will route to NSDB. The OPT CODE field indicates these circuits. Message trunks are represented by M, Carrier circuits are represented by C, and Special services circuits are represented by S. To add any of these to the table, type the character indicating the type of service desired in the OPT CODE field. For the NMA/Switch flow, an M entry is required. Press the PF4 function key to ADD the service type.
4. You can list all service types in the table by entering EDIIS OPTION CD in the TABLE NAME field, NSDB in the TABLE KEY field, SCAN in the COMMAND field, and pressing the PF1 function key. All of the selected service types will appear in the OPT CODE field.
5. You can perform a FIND on a single service type by entering EDIIS OPTION CD in the TABLE NAME field, NSDB in the TABLE KEY field, the letter representing the service type (M, C or S) in the TABLE RECORD KEY field, and pressing the PF1 function key. If the service type is valid, (one that EDIIS routes to NSDB), the returned screen will display that service type in the OPT CODE field.

### 9.2.4 EDIIS UCT

**INTERFACE ID = TIRKS-NSDB-ITS/Digital**
**= TIRKS-NSDB-NMA**
**= TIRKS-NSDB (For MPI Index Page data)**

This TTS table defines to EDIIS the exception notice printer LTERM and sizing information for the input/output and TCM Parser and Mapper (TPAM) buffers. This table is preloaded and set up by the EDIIS System Administrator (refer to the *EDIIS Installation Guide*, BR 756-586-001, for additional information).

```
COMMAND          ***  TIRKS-TTS DATA SCREEN  ***                    /FOR


TABLE NAME:  EDIIS UCT          TABLE KEY:                      ADMIN AREA:
TABLE RECORD KEY:                                  # OF RECORDS:
NOTE: CONTAINS LTERM AND TPAM BUFFER SIZE       REL/LEV:            MOD:
   FIELD     FIELD
   NAME      VALUE
   EXLTERM           *
   TPAM BUF     *
   I/O BUFR     *
   MSGLEVEL  *
   RECOVERY  *
   SYSDUMP   *









TTS210I FIND COMPLETED
```

**Figure 9–4.** EDIIS UCT Table

VERIFICATION OF THE EDIIS UCT TABLE

Follow the steps below to verify this table:

1. Type /FOR ZRTDSO and press the ENTER key.
2. On the returned screen, type EDIIS UCT in the TABLE NAME field. Perform a FIND by pressing the PF1 function key.
3. Verify that the following fields are populated as shown:

   TPAM: 040
   I/O BUFR: 200
   EXLTERM: the appropriate LTERM ID for exception notices.
   This is an 8–character alphanumeric field.

## 9.3 The SOAC Tables Required by NSDB

SOAC documents that may be useful to the NSDB System Administrator:

- *FACS Service Order Table Operations Performance Aid* (BR 753–303–217)
- *FACS Service Order Table User Guide* (BR 753–303–218)
- *FACS SOAC/NSDB Interface Performance Aid* (BR 753–303–440)

This section is intended to give the NSDB System Administrator an overview of the BCC– tuneable SOAC tables that are used to filter the service order and work order data that is sent to NSDB. The responsibility for the administration of these tables falls to the SOAC System Administrator.

16 of the SOAC tables described in this Guide are used in the selection and routing of service order and work order data to NSDB. These tables are briefly described in the following sections. There are other tables (not described in this document) that also influence the SOAC-NSDB interface. These tables are Bellcore–maintained, and are specific to each BCC region.

The SOAC tables that influence NSDB can be grouped into six functional areas of data selection and routing. Some tables are used in more than one functional area; the tables described in this Guide that are used in this way are highlighted.

- Determining NSDB involvement in service orders

  **Service Order Parameter Table**
  **Operations Wire Center Table**
  **Service Order Type Table**
  **Service Order Section Table**
  **USOC Table**
  **FID Data Table**
  **NSDB Options Table**

- Creation of data items needed by NSDB

  **FID Data Table**
  **USOC Table**
  **Exchange Table**
  **CLLI Code Table**
  **Fixed Identification Section Table**

- Generation and transmittal of TIRKS tag data

  **FID Data Table**
  **USOC Table**
  Supergrouping Table (not described in this document)
  TFS CLLI Code Table (not described in this document)
  TFS Option Table (not described in this document)
  Service Code Data Table (not described in this document)
  **TFS Circuit ID Hierarchy Table**
  TGAR Parse Rules (not described in this document)

- Routing of messages to NSDB

  **Output Routing Table**
  **Position ID Device Type Table**
  **Device Table**

- Routing of notifiers to an NSDB printer

  **Output Routing Table**
  Exception Code Analysis Table (not described in this document)
  **Position ID Device Type Table**
  **Device Table**
  Work Center Table (not described in this document)

- Routing of work order messages to NSDB

  **Common Update Wire Center Status Table**
  Parsing Rules (not described in this document)
  CT-AR Table (not described in this document)
  NSDB-AR Table (not described in this document)
  NSDB-MAP Table (not described in this document)
  Exception Code Analysis Table (not described in this document)

### 9.3.1 CLLI Code Table

This table is used to translate FACS internal wire center identifiers to the equivalent CLLI codes. All FACS wire center codes and associated CLLI codes should appear on this table.

### 9.3.2 Common Update Wire Center Status Table

This table is used to determine whether or not work order messages should be sent to an NSDB-involved wire center. This table also identifies groupings of NSDB-involved wire centers for reports purposes.

### 9.3.3 Destination Work Center Table

This table is used to associate "logical destinations" with NSDB "work centers".

### 9.3.4 Device Table

This table is used to identify the output characteristics of the devices identified in the Output Routing Table and the Position Identifier Device Type Table.

### 9.3.5 Exchange Table

This table is used to identify the Operations Administrative Area (OPSAA) associated with each BCC exchange. This OPSAA data is needed by NSDB to build the appropriate Circuit Layout Order (CLO).

### 9.3.6 FID Data Table

This table is used to identify the Field Identifiers (FIDs) that require Manual Assist Planning messages to be sent to NSDB. It is also used to determine the processing that should take place in the event of a Parsing Rule Failure (PRF).

### 9.3.7 Fixed ID Table

This table is used by SOAC to identify the location of the FID data in the Fielded Identification Section of the service order. Although there are no FIDs that are uniquely required by NSDB on this table, there are FIDs that are sent to NSDB which must be identified. These FIDs are:

- For POTS and nondesigned specials:

  CS
  DD
  SLS
  TN

- If POTS, nondesigned specials, and designed specials are processed:

  APP
  CS
  DD
  SLS
  TN

### 9.3.8 NSDB Options Table

This table is used to determine whether or not certain criteria are to be used to select the candidate circuits to send to NSDB. The selection of circuits by MCN FID, ISDN, or COLAN is specified on this table.

### 9.3.9 Operations Wire Center Table

This table is used to identify the individual SOAC wire centers that are used to send messages to NSDB (under the appropriate conditions). It also identifies by SEC the NSDB machine that will receive the messages and any error notifiers.

### 9.3.10 Output Routing Table

This table is used to route service order and work order messages to the desired NSDB machine. The NSDB SEC code is used to identify the appropriate NSDB copy. The Output Routing Table is also used to route error notifiers to the desired NSDB machine. A default routing entry is provided.

### 9.3.11 Position Identifier Device Type Table

This table is used to associate input devices with output devices. The NSDB System Administrator's terminal and printer are identified by entries made on this table, and are associated with corresponding devices in the SOAC system.

### 9.3.12 Service Order Parameter Table

This table is used to turn the SOAC-NSDB interface "on" or "off". It also identifies by System Entity Code (SEC) the SOAC machine that is used to send messages to NSDB if the interface is "on".

### 9.3.13 Service Order Section Table

This table is used to determine whether NSDB should receive a Manual Assist message based on a service order section.

### 9.3.14 Service Order Type Table

This table is used to define the service order types that are to be processed by SOAC for the SOAC-NSDB interface. It also is used to identify the types of service orders for which SOAC should create Manual Assistance (MA) precompletion assignment messages.

### 9.3.15 TFS Circuit ID Hierarchy Table

This table is used to generate a CLCI and determine the format of the circuit ID for designed special services. The circuit ID formats that are determined from this table are:

- Message Circuits (M)
- Serial Numbered Circuits (S)
- Telephone Numbered Circuits (T)

  These can be of 3 kinds:
  Telephone Number Circuit ID – Common Language
  Telephone Number Circuit ID – Non-Common Language
  Serial Number Circuit ID – Non-Common Language

### 9.3.16 USOC Table

This table is used to determine NSDB involvement based on class of service. It is also used to identify COLAN and ISDN service, and to identify circuit terminations on the service order.

## 9.4 TCM Online Formats

Four TCM online formats are described in this section. These screens are the Message Administration screen (VJMPMSG), the Test Message Generator screen (VJMPTST), the Network Administration screen (VJMPNET), and the Network Statistics screen (VJMPSTS). The purpose of this section is to give the NSDB System Administrator an introduction to these screens as they relate to the functions of a system administrator. For more detailed information about the screens, refer to the following Bellcore documents:

- *Operations Systems Interface Administrator's Guide*

  BR 190–518–001
  This document leads the system administrator through the procedures required to set up and troubleshoot the interfaces currently supported by the WFA and NSDB product lines.

  It is particularly recommended to the NSDB system administrator because it takes a practical, step–by–step approach to the installation and maintenance of NSDB interfaces. It helps the system administrator to:

  — Acquire a general understanding of the WFA and NSDB interface structure

  — Acquire a general understanding of the TIRKS Communication Module (TCM) software

  — Display, verify, and populate the Operations and Provisioning Bellcore TIRKS Table System (TTS) tables

  — Display and populate TCM formats

  — Create and verify the WFA and NSDB system interface communication paths

  — Display and populate the Operations Interface (OI) system formats for establishing terminal emulation

  — Display and use TCM formats for message processing and error resolution

  — Display the Open Query System (OQS) utilities for querying TCM databases.

- *TCM Overview*

  BR 756–573–301
  This document gives a "big picture" of the overall TCM system, and explains the purpose of each of the components.

- *TCM Routing Administration (RA) User Manual*

  BR 756–573–302
  This document describes the "heart of TCM", the Routing Administration component (RA). Detailed discussions are provided for Input Header, Class 1 and Class 3 Message, Acknowledgement, Error, and Dequeue processing.

- *TCM Message Administration (MA) User Manual*

  BR 756–573–303
  This document describes the tools used to monitor and correct messages routed through TCM. The VJMPMSG and VJMPTST formats are fully described.

- *TCM Network Administration (NA) User Manual*

  BR 756–573–304
  This document describes the tools used to monitor and maintain the network of communication channels that connect to TCM. The VJMPNET and VJMPSTS formats are discussed here as well.

- *TCM Translation Administration (TA) User Manual*

  BR 756–573–305
  This document describes the tools available to translate data that is incoming to, or outgoing from TCM. TCM translation rules are discussed in detail.

### 9.4.1 Message Administration Screen (VJMPMSG)

This screen is used to display and correct messages that are held in the TCM TLOG database. (The TLOG database is where the message failures from the TCM Parser and Mapper (TPAM) are stored.) In addition, corrected messages can be resubmitted for processing from this screen.

```
 *  NSDB-TCM:  MESSAGE ADMINISTRATION  (VJMPMSG) *      /FOR            PAGE    1
COMMAND =>                                              88.315 DATE MM/DD/YY
ACNO                              I/O     MSG STATUS              TIME 18:18:01
ERROR CODE                  OUTPUT MSG KEY              SEARCH AGR NAME
SOURCE (TSYS)               TIME STAMP                  TMST (END)
TARGET (RSYS)               PATHID           SCTYPE
ERROR
NOTE :
```

| FIND | FWD | BACK | | UPD | NEXT | SAVE | RFSH | RSND | DELET | HELP | PRNT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PF1 | PF2 | PF3 | PF4 | PF5 | PF6 | PF7 | PF8 | PF9 | PF10 | PF11 | PF12 |
| PF13 | PF14 | PF15 | PF16 | PF17 | PF18 | PF19 | PF20 | PF21 | PF22 | PF23 | |

For complete information about the use of VJMPMSG and several other associated Message Administration screens, refer to Bellcore document BR 756–573–303, the *TCM Message Administration (MA) User Manual.*

## 9.4.2 Test Message Generator Screen (VJMPTST)

This screen is used to build test messages to simulate transmission of messages to NSDB from another system, or vice versa. It is useful for testing the communications paths for an external system before it is turned up.

```
COMMAND:          NSDB-TCM:  TEST MESSAGE GENERATOR  (VJMPTST)          /FOR

TEST MESSAGE ID:                    IMS TRANCODE:                  PAGE NUMBER:









1/13 FIND         2/14 FORWARD        3/15 BACK         4/16 ADD       5/17 UPDATE
8/20 REFRESH      9/21 SEND           10/22 DELETE      11/23 HELP     12/24 PRINT
```

| FIND | FWD | BACK | ADD | UPD | | | RFSH | RSND | DELT | HELP | PRNT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PF1 | PF2 | PF3 | PF4 | PF5 | PF6 | PF7 | PF8 | PF9 | PF10 | PF11 | PF12 |
| PF13 | PF14 | PF15 | PF16 | PF17 | PF18 | PF19 | PF20 | PF21 | PF22 | PF23 | PF24 |

For complete information about the use of VJMPTST and several other associated Message Administration screens, refer to Bellcore document BR 756–573–303, the *TCM Message Administration (MA) User Manual.*

### 9.4.3 Network Administration Screen (VJMPNET)

This screen is used to establish and display the communications paths between NSDB and the supported systems.

```
*  NSDB-TCM: NETWORK ADMINISTRATION (VJMPNET) *      /FOR            PAGE
SEC:                PATHID:              SCENARIO TYPE:            DATE
                                                                   TIME
********** SEC  INFORMATION  **********
SEC TYPE:   RELEASE LEVEL:
LINK STATUS     (TO) MAY-I:       (FROM) MAY-I:
PROCESSING LEVEL   RESEND:               HOLD:
DESCRP:

********* PATH  INFORMATION  **********
                                        (MYSEC)
    MESSAGE CLASS:                    RULE ID:
        LINK TYPE:                  LINK NAME:
LINK STATUS MAY-I:            LINK STATUS CAN-I:
   DEQUEUE COUNT:                   INPUT MFD:
        TRANCODE:                  OUTPUT MFD:
 TCM ERROR LDEST:               APP ERROR LDEST:

   *** LTERMS FOR EXCEPTION NOTICES ***
   PRIMARY:              SECONDARY:              REMOTE NOTIFICATION:

1/13 FIND         2/14 NEXT PATH       4/16 ADD       5/17 UPDATE   6/18 NEXT SEC
7/19 VJMPERR      8/20 REFRESH         9/21 VJMPSTS 10/22 DELETE  11/23 HELP
```

| FIND | FWD | | ADD | UPD | NEXT | VJMPERR | RFSH | VJMPSTS | DELT | HELP | PRNT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PF1 | PF2 | PF3 | 4 | PF5 | PF6 | PF7 | PF8 | PF9 | PF10 | PF11 | PF12 |
| PF13 | PF14 | PF15 | PF16 | PF17 | PF18 | PF19 | PF20 | PF21 | PF22 | PF23 | |

For more information about the VJMPNET entries required to establish the specific communications paths used by NSDB, refer to the *Operations Systems Interface Administrator's Guide* (BR 190–518–001). For a detailed discussion about the VJMPNET format and all its features, refer to Bellcore document BR 756–573–304, the *TCM Network Administration (NA) User Manual.*

### 9.4.4 Network Statistics Screen (VJMPSTS)

This screen is used to display and monitor the volume and status of messages processed by TCM for today and yesterday. This screen is useful because it enables the NSDB System Administrator to see on a single screen the number of total messages compared to those that have failed. By periodically viewing this screen, the TCM network can be easily monitored.

```
 *  NSDB-TCM:  NETWORK MSG VOLUMES      (VJMPSTS) * /FOR           PAGE    1
SEC:                   PATHID:                SCENARIO TYPE:  88.315 DATE MM/DD/YY
                             CURRENT MESSAGE STATUS                    TIME 15:22:46
DEFERRED:                      TCM  :                    TRNSLTN:
HELD                           PARSE:                    APPLCTN:
PEND ACK:                      MAP  :                    RETURND:

TOTAL MESSAGE VOLUMES        YESTERDAY               TODAY
                     INPUT:
                 DELIVERED:
                  DEFERRED:
                      HELD:
              ACKNOWLEDGED:
           RESENT TO MYSEC:
           RESENT BY MYSEC:
                TCM ERRORS:
             PARSER ERRORS:
             MAPPER ERRORS:
       TRANSLATION ERRORS:
       APPLICATION ERRORS:
     APPLICATION WARNINGS:
        RETURNED TO MYSEC:
        RETURNED BY MYSEC:
```

| FIND | | | | | | | RFSH | VJMPNET | | HELP | PRNT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PF1 | PF2 | PF3 | PF4 | PF5 | PF6 | PF7 | PF8 | PF9 | PF10 | PF11 | PF12 |
| PF13 | PF14 | PF15 | PF16 | PF17 | PF18 | PF19 | PF20 | PF21 | PF22 | PF23 | |

For a detailed discussion about the VJMPSTS format and all its features, refer to Bellcore document BR 756–573–304, the *TCM Network Administration (NA) User Manual.*

## 9.5 Open Query System (OQS)

OQS is used to generate user–customized and predefined reports. The NSDB System Administrator can use OQS in two areas of NSDB administration: TCM message administration and queries of the NSDB databases.

The ability to query the TCM Message Administration database is important because it reduces the time required to identify and fix messages that are in error, held, or deferred. This feature also eliminates the possibility of inducing errors into the NSDB databases while using the Database Modifier for the purpose of message administration.

The ability to query the nondesigned databases enables the NSDB System Administrator to create customized views of data based on various criteria. The nondesigned databases that can be queried using OQS are the Line Record database, Set Services database, Order Activity database, Hunt Group database, and the Alias database. The designed databases that can be queried are the In-Effect Circuit/Carrier database, Pending Circuit/Carrier database, Message database, Multi-Point database, and Circuit Index databases.

The new DB2 databases, Measurement and Facility–to–Circuit, are also designed to accommodate OQS queries. The TTS table OQS DB2 TABLES maps the OQS database name to the actual DB2 table name, and includes the high–order node of the DB2 table (OWNERID) to permit SQL statements to be built.

In order to use OQS for NSDB, S1 security must be established and the authorized OQS users must be defined in the OQS USERID table. For complete security procedures, see the *NSDB Security Administrator's Guide*, BR 190-534-003. For procedures describing the use of OQS, see the *NSDB OQS User Manual*, BR 190-534-300. To set up the OQS USERIDS table, refer to chapter 7 of this Guide.

## 9.6 How NSDB Derives a Line Record from the TIRKS Circuit Detail

The purpose of this section is to describe the method by which NSDB derives loop data from TIRKS Circuit Detail information. The following steps are taken:

1. NSDB looks for an indication of an A end by checking to see whether any of the three following tags are populated in the CKTHEADR aggregate of the FCIF message:

- LOOP#A
- LCONNAMA
- LCONADRA

NSDB interprets the existence of data in any one of these fields as a signal to add an A end circuit into the Line Record database.

2. NSDB then looks for an indication of a Z end by checking to see whether any of the three following tags are populated in the CKTHEADR aggregate of the FCIF message:

- LOOP#Z
- LCONNAMZ
- LCONADRZ

NSDB interprets the existence of data in any one of these fields as a signal to add a Z end circuit into the Line Record database.

3. If NSDB did not find data in any of the six fields listed above, then a Z–end line record is created.
4. After NSDB has an indication of an A end, a Z end , or both, it checks the CDCABLES aggregate to determine whether loop data exists. (Non–blank values signify the existence of loop data.)
5. If loop data exists, NSDB then checks the LFACTYPE tag in the CDCABLEE aggregate to learn whether the loop data represents A end or Z end information. This is accomplished by scanning the first and second characters of the ASGID tag.

- The first character indicates loop end. Values can be "A" or "Z".
- The second character defines circuit action. Values can be "A" (Add) or "D" (Disconnect). If the second character is defined as an "A", the loop data is stored in the Line Record database. If it is defined as a "D", the loop data is ignored, and NSDB continues to look for useable A end or Z end loop data.

6. The SVCMPT tag in the CDCABLES aggregate indicates the use of a pair. Valid values for this tag are listed below:

- X2 (two–wire)
- XR (receive pair of a four–wire)
- XT (transmit pair of a four–wire)

## 9.7 Procedure for Turning Up the MPI Feature

To turn up the MPI feature that became available with NSDB Release 3.1, a few steps must be taken prior to the TIRKS MPI load and prior to using the online interface from the TIRKS system. The following procedure outlines what must be done to the databases to prepare them for new TIRKS data. This will ensure that MPI data in NSDB will be managed and displayed correctly to the end user.

1. Initialize the MPI databases (VJDBM0DD through VJDBM9DD). This will clear out all of the old MPI records with "J" facility codes (those built by the internal NSDB software for digital circuits).
2. Run the Auditor/Corrector (run title – VJCDM05). This will delete the old index records in the VJDBX1DD index database (the index to VJDBM0–9DD).
3. Turn the "LOAD–IP" flag in the ODB CO OPTIONS TTS table to "Y". This will cause NSDB to stop building the internally–generated index pages (the J facility code) and will allow NSDB to accept the data received from the ISSUIX event in the TIRKS EDIIS table.
4. Extract and load the MPI data from the TIRKS system.
5. Turn on the EDIIS event ISSUIX in the TIRKS system.

You may decide to perform steps 4 and 5 in reverse order from that stated in the instructions above. Detailed information on the TTS tables used to activate this interface can be found in chapter 7 of this Guide. Information describing how to extract MPI circuits can be found in the TIRKS runbook for ZRCDX05.

## 9.8 Archive and Retrieval Capabilities for NSDB

### 9.8.1 Overview of the Archive and Retrieval Capabilities for NSDB

Archive/Retrieval functionality in NSDB permits the removal of data from online databases for storage (archive) on magnetic or cassette tape with the option to review this data (retrieval) at a later time. This functionality includes an interactive interface and three BMP (Batch Message Processing) processes. The interactive interface provides the BCCs with the ability to create, update, and delete selection requests for both archive and retrieval processes. The BMP processes are comprised of an archive run, a retrieval run, and a tape identification run.

Archive, retrieval, and tape identification BMP runs access source data from tables that have been "preset" by Bellcore with information supporting this feature. These tables are the Archive/Retrieval Archive File Descriptor (A/R AFD) TTS tables and the Archive/Retrieval Record Format Descriptor (A/R RFD) TTS tables. They are described in the section "TTS Tables Supporting Archive/Retrieval Functionality".

When the AFD and RFD tables are preset, certain fields are specified as "key fields". These fields become the keys for extracting archived data from tapes. The system requires users to select at least one key field when constructing a retrieval selection request in order to streamline processing functions and to avoid sequential searching of all tape records.

The tape identification BMP run uses the criteria defined for a retrieval run on a selection request to determine the corresponding tape volume-serial numbers. Executing the tape identification run previous to the retrieval run ensures that the tapes needed for the retrieval run are available and can be mounted on the tape drive for retrieval processes.

After the tape identification run is executed, the user can view the tape volume-serial numbers which will be needed for the retrieval run and the number of records that will be retrieved as a result. This information assists in the decision to either perform the run or modify the selection criteria defined for the run.

**NOTE –** While the Archive/Retrieval BMPs are running, the Archive/Retrieval screens are not available.

Information about output reports for these runs can be obtained by consulting the associated run books. Refer to the table below for details.

**Table 9–3.** Run Books Associated with Archive/Retrieval Processes

| Run Book Title: | BR Number: | Run Name: |
|---|---|---|
| *Archive Run* | BR 190-534-058 | VJARM01 |
| *Archived Data Retrieval Run* | BR 190-534-059 | VJARM02 |
| *Tape Identification Process* | BR 190-534-060 | VJARM03 |

If your company is currently using OPCDB at a 3.0 or 3.0.x release level and you plan to upgrade to an NSDB 3.1 level or higher, you will need to perform a Rebuild to take advantage of archive functionality. A Rebuild primes the VJDBA3DD partition (described in the section "Internal Processing in support of Archive Functionality") with new data received between NSDB Releases 3.0 and 3.1. Rebuilding can be accomplished either from the VJDBADM screen by using a REBUILD command with a BUILD ARCHIVE keyword, or via a BMP run.

#### 9.8.1.1 Archive Source Databases

The primary databases associated with Archive/Retrieval capabilities for NSDB are the Line Record database, the Service Order Image database, the Audit database and the designed circuit details databases (In–effect Circuit/Carrier, Circuit Index, and Message). The archive process also retrieves some data from the Alias database, the Location Reference database (LOCREF), and TTS tables ODB AUDIT, ODB IMAGE, and ODB CO OPTIONS. These tables are required to build the archive record in Alias, but they are not used by the VJARM01 process.

Records are stored, archived, and retrieved in system time. Time zone conversions are not currently supported in any of the processes associated with archive and retrieval functions.

### 9.8.1.2 Internal Processing in Support of Archive Functionality

The Alias database is comprised of three partitions: the VJDBA1DD and VJDBA2DD partitions contain aliases for line records, and the VJDBA3DD partition identifies the data to be archived. The VJDBA3DD partition, in turn, contains four records (IMS segments): the Archive Audit (AA) record, the Archive Image (AI) record, the Archive Line (AL) record, and the new Archive Design (AD) record created for Release 3.2. One or more of these records are stored in the VJDBA3DD partition for any given circuit at the time of completion.

> **NOTE –** The AL and AD records are stored at the time of completion for disconnects and cancelled circuits. Completed cancels are not archived; they are deleted when the Archive Run VJARM01 is executed. (In-effect circuits and Add, Change, or Rearrange orders are not archived in the AL and AD records.)
>
> The AI record is stored at the time of completion if the service order image is associated with a circuit that is an archive candidate. The AA record is stored at the time of completion whenever an archive candidate has audit information.

For a given circuit, "time of completion" corresponds to the NSDB circuit status. NSDB circuit status can be determined by viewing the *OPS/PRV STATUSES* field on the VJDBADM screen. "*OPS*" represents the TIRKS->WFA/C status and "*PRV*" represents the SOAC status, if the SOAC interface is active. If the SOAC interface is not turned up, the *PRV* status assumes the same value as the *OPS* status. Archive records will be stored in the Alias database when both the *OPS* status and the *PRV* status become in-effect. If either status displays as pending (**P**), NSDB will *not* consider the circuit complete.

The completion notification procedure described above constitutes the most common method of setting the archive date. There are two circumstances which can cause the archive date to differ from the date when NSDB is notified of circuit completion:

- When the Rebuild process is initially executed to prime the Archive feature, the archive date is set to the date of the Rebuild for any archived records completed before that time.
- If a circuit stored in the Archive Audit (AA) record is updated, the date of the update will replace the completion date, thus delaying the date that the circuit is archived. A REBUILD transaction from the VJDBADM screen for a circuit or a partition is also considered an update and will change the archive date for any circuit stored in the AA record. [This does not apply to Archive Image (AI), Archive Line (AL), or Archive Design (AD) records; updates to these records do not affect the archive date for stored circuits.]

### 9.8.1.3 Archive Modes

The next table explains alternative modes of archiving data by means of the generic archive and retrieval system. Archive mode is defined on the JCL (Job Control Language) for the archive run.

**Table 9–4.** Archive Modes

| Mode: | Description: |
|---|---|
| Production Mode | When executing a run in "production mode", the records are written to tape and then deleted from the database. On the JCL, the default run mode is Production (P). |
| Test Mode | "Test mode" permits the execution of the archive BMP run and a display of the candidate records for archive. It calls for writing eligible records to tape while retaining them in the source database. A new tape is used whenever test mode is selected in order to keep the tape free of unwanted data. Test mode does not off-load records. Records that have been archived in test mode can be viewed by executing a retrieval run in test mode. |
| Delete Mode | "Delete mode" deletes records from the database but does not write them to a tape. This mode should be used with *EXTREME* caution. ("D" = Delete) |

**NOTE –** Test mode record counts may differ from corresponding production mode record counts for the Service Order Image (SOI) database and the Audit database. Such inconsistencies will not occur after NSDB Release 3.1.1 for the SOI database and after NSDB Release 3.2 for the Audit database.

### 9.8.1.4 Archive Selection Criteria

The primary criteria for archiving data is an aging factor. The initial archive run with an age factor is usually executed when the BCC database manager determines that a database is nearing capacity. This factor will continue to be applied so as to maintain a predetermined percentage of capacity.

The archive process accesses aging criteria via the Archive Control Card (VJARCNTL) screen, which is described in the section "Online Screens Supporting Archive/Retrieval Functionality". The age factor is typically set to fall within a range of 30 to 90 days, depending on the daily volume of records. A minimum factor of 30 days has been established for normal work operations. If a value of less than 30 days is entered by the user, it will be automatically converted by the system to a value of 30 days. The 30 day limitation can be overridden for the NSDB product by appending a special command to the age factor.

If the aging factor alone proves to be insufficient for archiving needs, an option to archive by special selection criteria using the Archive Selection Display (VJARDSPA) screen can be utilized. This is a more complex procedure, and should be attempted only by users with a strong knowledge of field attributes and a clear view of the characteristics desired for a particular archive run.

### 9.8.2 TTS Tables Supporting Archive/Retrieval Functionality

The primary databases associated with Archive/Retrieval capabilities for NSDB are the Line Record database, the Service Order Image database, the Audit database, and the designed circuit details databases (In–effect Circuit/Carrier, Circuit Index, and Message). Data extracted from these databases are defined in the Archive/Retrieval Archive File Descriptor (A/R AFD) TTS tables and the Archive/Retrieval Record Format Descriptor (A/R RFD) TTS tables. The archive process also retrieves some data from the Alias database, the Location Reference database (LOCREF), and TTS tables ODB AUDIT, ODB IMAGE, and ODB CO OPTIONS. These tables are required to build the archive record in Alias, but they are not used by the VJARM01 process.

The AFD tables are used to define the logical relationship between archive records. In addition to defining the archive run name, AFD tables contain Record Format Descriptor (RFD) names which point to RFD tables that define the data to be extracted on archive runs.

The RFD tables define the key and non-key fields within target records that will be extracted on the archive run. The distinction between these two types of fields is important. Key fields are used to create retrieval indexes which allow for a more efficient means of processing than simple sequential searches of all tape records. For this reason, the system requires users to select at least one key field when constructing a retrieval selection request.

> **NOTE –** The AFD and RFD tables are preset by Bellcore and *cannot be modified in the field.* The data on these tables can be viewed using standard TTS procedures (e.g., after a FIND, the user can scroll forward through multiple records via the PF6 function key).

### 9.8.2.1 Accessing A/R AFD and A/R RFD TTS Tables

The AFD and RFD TTS tables are obtained via the standard TTS format "VJTDSO". An Archive Name is used to access an AFD or RFD table whenever there is a need to review information for an archive run. For NSDB, there are four archive runs which correlate to the primary databases (Audit, Service Order Image, Line Record, and the designed circuit details databases). The Archive Names associated with each of these runs are listed below.

**Table 9–5.** Valid Archive Names for NSDB

| Valid Archive Name: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

The Archive Name is also used as a table key to access the AFD TTS tables. For RFD TTS tables, this name is concatenated with the AFD table record keys to produce RFD table keys.

### 9.8.2.2 Displaying A/R AFD TTS Tables

**Table 9–6.** Criteria for the Display of A/R AFD TTS Tables

| Table Name: | Table Key: | Table Record Key: |
|---|---|---|
| A/R AFD | VJDBAAU1 | A000 |
| A/R AFD | VJDBAAU1 | A1HD |
| A/R AFD | VJDBAAU1 | A2LG |
| A/R AFD | VJDBADN1 | D000 |
| A/R AFD | VJDBADN1 | D1HD |
| A/R AFD | VJDBADN1 | D2A |
| A/R AFD | VJDBADN1 | D2B |
| A/R AFD | VJDBADN1 | D2C |
| A/R AFD | VJDBADN1 | D2D |
| A/R AFD | VJDBADN1 | D2E |
| A/R AFD | VJDBADN1 | D2I |
| A/R AFD | VJDBADN1 | D2IH |
| A/R AFD | VJDBADN1 | D2L |
| A/R AFD | VJDBADN1 | D2M |
| A/R AFD | VJDBADN1 | D2MH |
| A/R AFD | VJDBADN1 | D2N |
| A/R AFD | VJDBADN1 | D2O |
| A/R AFD | VJDBADN1 | D2P |
| A/R AFD | VJDBADN1 | D2PH |
| A/R AFD | VJDBADN1 | D2Q |
| A/R AFD | VJDBADN1 | D2R |
| A/R AFD | VJDBADN1 | D2S |
| A/R AFD | VJDBADN1 | D2SA |

**Table 9–6.** Criteria for the Display of of A/R AFD TTS Tables (continued)

| Table Name: | Table Key: | Table Record Key: |
|---|---|---|
| A/R AFD | VJDBADN1 | D2SZ |
| A/R AFD | VJDBADN1 | D2T |
| A/R AFD | VJDBADN1 | D2TD |
| A/R AFD | VJDBADN1 | D2TP |
| A/R AFD | VJDBADN1 | D2V |
| A/R AFD | VJDBADN1 | D2Y |
| A/R AFD | VJDBADN1 | D2Z |
| A/R AFD | VJDBAIM1 | I000 |
| A/R AFD | VJDBAIM1 | I1GH |
| A/R AFD | VJDBAIM1 | I2GC |
| A/R AFD | VJDBALR1 | L000 |
| A/R AFD | VJDBALR1 | L1HD |
| A/R AFD | VJDBALR1 | L2FN |
| A/R AFD | VJDBALR1 | L2HT |
| A/R AFD | VJDBALR1 | L2RM |
| A/R AFD | VJDBALR1 | L2SD |
| A/R AFD | VJDBALR1 | L2US |
| A/R AFD | VJDBALR1 | L3TF |

### 9.8.2.3 Sample A/R AFD TTS Tables

The figure below displays an A/R AFD table record with a table record key of L000. This AFD table record performs administrative functions such as defining the appropriate version of the AFD/RFD tables and providing linkage to the desired application (i.e., NSDB).

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  A/R AFD           TABLE KEY: VJDBALR1           ADMIN AREA:
TABLE RECORD KEY:  L000                          # OF RECORDS: 0008
NOTE: A/R ARCHIVE FILE DESCRIPTORS TABLE       REL/LEV: 3.2        MOD: N
   FIELD    FIELD
   NAME     VALUE
   RFDNAME  L000*
   ARCHDESC LINE RECORD CTL REC *
   PARENT   001L*
   SOURCE   VJDBAPV *









TTS210I FIND COMPLETED
```

**Figure 9–5.** A/R AFD TTS Table with Table Record Key of L000

The figure below displays an A/R AFD table record with a table record key of L1HD. This AFD record points to RFD tables which specify some of the data fields to be archived.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  A/R AFD           TABLE KEY: VJDBALR1           ADMIN AREA:
TABLE RECORD KEY:  L1HD                          # OF RECORDS: 0008
NOTE: A/R ARCHIVE FILE DESCRIPTORS TABLE       REL/LEV: 3.2       MOD: N
   FIELD    FIELD
   NAME     VALUE
   RFDNAME  L1HD*
   ARCHDESC LINE RECORD HEADER  *
   PARENT       *
   SOURCE           *









TTS210I FIND COMPLETED
```

**Figure 9–6.** A/R AFD TTS Table Record Key of L1HD

The figure below displays an A/R AFD table record with a table record key of L2FN. This AFD record points to the RFD tables specifying additional fields associated with the Line Record database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  A/R AFD           TABLE KEY: VJDBALR1           ADMIN AREA:
TABLE RECORD KEY:  L2FN                          # OF RECORDS: 0008
NOTE: A/R ARCHIVE FILE DESCRIPTORS TABLE       REL/LEV: 3.2       MOD: N
   FIELD     FIELD
   NAME      VALUE
   RFDNAME   L2FN*
   ARCHDESC  LINE RECORD FACILITY*
   PARENT    L1HD*
   SOURCE             *









TTS210I FIND COMPLETED
```

**Figure 9–7.** A/R AFD TTS Table with Table Record Key of L2FN

The figure below displays an A/R AFD table record with a table record key of L3TF. This AFD record identifies other RFD tables that define additional data from the Line Record database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  A/R AFD           TABLE KEY: VJDBALR1           ADMIN AREA:
TABLE RECORD KEY:  L3TF                        # OF RECORDS: 0008
NOTE: A/R ARCHIVE FILE DESCRIPTORS TABLE      REL/LEV: 3.2       MOD: N
   FIELD    FIELD
   NAME     VALUE
   RFDNAME  L3TF*
   ARCHDESC LINE RECORD TRANSPOR*
   PARENT   L2FN*
   SOURCE           *









TTS210I FIND COMPLETED
```

**Figure 9–8.** A/R AFD TTS Table with Table Record Key of L3TF

## Field Descriptions

### RFDNAME

| | |
|---|---|
| Full Name: | Record Format Descriptor Name |
| Definition: | The name of the record format descriptor being described. |
| Data Content: | 4 A/N |

### ARCHDESC

| | |
|---|---|
| Full Name: | Archive Description |
| Definition: | The description of this archive record segment. |
| Data Content: | 20 A/N |

### PARENT

| | |
|---|---|
| Full Name: | Parent |
| Definition: | The name of the parent of this segment. |
| Data Content: | 4 A/N |

### SOURCE

| | |
|---|---|
| Full Name: | Source |
| Definition: | The name of the application program supplying this data. |
| Data Content: | 8 A/N |

### 9.8.2.4 Displaying A/R RFD TTS Tables

**Table 9–7.** Criteria for Display of A/R RFD TTS Tables

| Table Name: | Table Key: | Table Record Key: |
|---|---|---|
| A/R RFD | VJDBAAU1A1HD | NOTE * |
| A/R RFD | VJDBAAU1A1LG | NOTE * |
| A/R RFD | VJDBAAU1A2LG | NOTE * |
| A/R RFD | VJDBADN1D1HD | NOTE * |
| A/R RFD | VJDBADN1D2A | NOTE * |
| A/R RFD | VJDBADN1D2B | NOTE * |
| A/R RFD | VJDBADN1D2C | NOTE * |
| A/R RFD | VJDBADN1D2D | NOTE * |
| A/R RFD | VJDBADN1D2E | NOTE * |
| A/R RFD | VJDBADN1D2I | NOTE * |
| A/R RFD | VJDBADN1D2IH | NOTE * |
| A/R RFD | VJDBADN1D2L | NOTE * |
| A/R RFD | VJDBADN1D2M | NOTE * |
| A/R RFD | VJDBADN1D2MH | NOTE * |
| A/R RFD | VJDBADN1D2N | NOTE * |
| A/R RFD | VJDBADN1D2O | NOTE * |
| A/R RFD | VJDBADN1D2P | NOTE * |
| A/R RFD | VJDBADN1D2PH | NOTE * |
| A/R RFD | VJDBADN1D2Q | NOTE * |
| A/R RFD | VJDBADN1D2R | NOTE * |
| A/R RFD | VJDBADN1D2S | NOTE * |
| A/R RFD | VJDBADN1D2SA | NOTE * |
| A/R RFD | VJDBADN1D2SZ | NOTE * |
| A/R RFD | VJDBADN1D2T | NOTE * |
| A/R RFD | VJDBADN1D2TD | NOTE * |
| A/R RFD | VJDBADN1D2TP | NOTE * |

**Table 9–7.** Criteria for Display of A/R RFD TTS Tables (continued)

| Table Name: | Table Key: | Table Record Key: |
|---|---|---|
| A/R RFD | VJDBADN1D2V | NOTE * |
| A/R RFD | VJDBADN1D2X | NOTE * |
| A/R RFD | VJDBADN1D2Y | NOTE * |
| A/R RFD | VJDBADN1D2Z | NOTE * |
| A/R RFD | VJDBAIM1I1GH | NOTE * |
| A/R RFD | VJDBAIM1I2GC | NOTE * |
| A/R RFD | VJDBALR1L1HD | NOTE * |
| A/R RFD | VJDBALR1L2FN | NOTE * |
| A/R RFD | VJDBALR1L2HT | NOTE * |
| A/R RFD | VJDBALR1L2RM | NOTE * |
| A/R RFD | VJDBALR1L2SD | NOTE * |
| A/R RFD | VJDBALR1L2US | NOTE * |
| A/R RFD | VJDBALR1L3TF | NOTE * |

NOTE * There is one table record key for each screen FID and each hidden field in the database. The table record key carries the format "abbccddddddddddddd" where,

- "a" is the value of the field KEYFLAG, which can be **K** (Key), **N** (Non-key), or **S** (non-key Structure)
- "bb" is the value of the field FLDGRP, which can be a number in the range of 00 through 99
- "cc" is the value of the field FLDSEQ, which can be a number in the range of 00 through 99
- and "dddddddddddd" is the value of the field designated as FLDNAME, which is the name given to the screen FID or the database field identifier. It is used to construct selection criteria for archive or retrieval requests. These values are equivalent to the FIELD value entries listed on the VJARDSPA, VJARDSPR, and VJARFNSD formats.

Table record keys for the RFD table can be displayed by executing the SCAN command on the TTS table format.

### 9.8.2.5 Sample A/R RFD TTS Tables

The figure below displays an A/R RFD table entry with a table key of VJDBALR1L1HD. The sample shown below defines the fields associated with the "Circuit Access Code" field from the Line Record database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  A/R RFD           TABLE KEY: VJDBALR1L1HD       ADMIN AREA:
TABLE RECORD KEY:  K9904CAC                    # OF RECORDS: 0233
NOTE: A/R RECORD FORMAT DESCRIPTORS TABLE      REL/LEV: 3.2       MOD: N
   FIELD    FIELD
   NAME     VALUE
   KEYFLAG  K*
   FLDGRP   99*
   FLDSEQ   04*
   FLDNAME  CAC          *
   FLDDESC  CIRCUIT ACCESS CODE *
   FLDSTR   00066*
   FLDLTH   008*
   SRCSTR   00119*
   SRCLTH   008*
   TYPE     C*
   JUST     L*
   FILL      *
   TRANS            *


TTS210I FIND COMPLETED
```

**Figure 9–9.** A/R RFD TTS Table Key of VJDBLE1L1HD

The figure below displays an A/R RFD table entry with a table key of VJDBALR1L1HD. The sample shown below defines the fields associated with the "Circuit Identification" field from the Line Record database.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR

TABLE NAME:  A/R RFD           TABLE KEY: VJDBALR1L1HD       ADMIN AREA:
TABLE RECORD KEY:  K9901CID                    # OF RECORDS: 0233
NOTE: A/R RECORD FORMAT DESCRIPTORS TABLE      REL/LEV: 3.2       MOD: N
   FIELD    FIELD
   NAME     VALUE
   KEYFLAG  K*
   FLDGRP   99*
   FLDSEQ   01*
   FLDNAME  CID          *
   FLDDESC  CIRCUIT ID           *
   FLDSTR   00001*
   FLDLTH   046*
   SRCSTR   00186*
   SRCLTH   046*
   TYPE     C*
   JUST     L*
   FILL      *
   TRANS            *


TTS210I FIND COMPLETED
```

**Figure 9–10.** A/R RFD TTS Table Key of VJDBALR1L1HD

## Field Descriptions

### KEYFLAG

Full Name: Key Flag

Definition: The flag identifying the type of field (key, non–key, or structure).

Data Content: 1 A

### FLDGRP

Full Name: Field Grouping Identifier

Definition: The number used to keep the elements of a structure together.

Data Content: 2 N

### FLDSEQ

Full Name: Field Sequence Number

Definition: The number used to control the sequencing of elements.

Data Content: 2 N

### FLDNAME

Full Name: Field Name

Definition: The name used to identify a field on screens and reports.

Data Content: 12 A/N

### FLDDESC

Full Name: Field Description

Definition: The field description printed on generic reports.

Data Content: 20 A/N

### FLDSTR

Full Name: Field Start

Definition: The starting location of the field in the archive buffer.

Data Content: 5 N

### FLDLTH

Full Name: Field Length

Definition: The length of the field in the archive buffer.

Data Content: 3 N

### SRCSTR

Full Name: Source Start

Definition: The starting location of the field in the application buffer.

Data Content: 5 N

### SRCLTH

Full Name: Source Length

Definition: The length of the field in the application buffer.

Data Content: 3 N

### TYPE

Full Name: Data Type

Definition: The data type of the field.

Data Content: 1 A

### JUST

Full Name: Justification

Definition: If field lengths are not equal, this determines how the data should be adjusted.

Data Content: 1 A

### FILL

Full Name: Fill Character

Definition: This value will serve as the fill character when field lengths are not equal.

Data Content: 1 A/N

## TRANS

| | |
|---|---|
| Full Name: | Translator Program |
| Definition: | The name of the program to be used for any special translations. |
| Data Content: | 8 A/N |

### 9.8.3 Online Screens Supporting Archive/Retrieval Functionality

Archive/Retrieval functionality is supported by eight online screens, each of which is discussed in detail in this section. A list of these screens follows:

- VJARMENU (Archive/Retrieval Menu screen)
- VJARCNTL (Archive Control Card screen)
- VJARDSPA (Archive Selection Display screen)
- VJARFNSD (Fieldname Selection Display screen)
- VJARLSTA (Archive Selection List screen)
- VJARDSPR (Retrieval Selection Display screen)
- VJARLSTR (Retrieval Selection List screen)
- VJARTLST (Archive/Retrieval Tapelist screen)

The Archive and Retrieval screens are controlled by S1 Security. You must use the VJS1SIGN screen to sign on in order to use the screens on the VJARMENU. If you are not authorized to use the Archive/Retrieval screens (via S1 Security for SYBSYS = ARR and GRID = FUNCTION) or have not signed on via VJS1SIGN, the system returns an error message. If the system displays the *USER MUST SIGN ON* message, display the VJS1SIGN screen and enter your signon user ID and your eight–character password. If you fail to sign on after three attempts, the system "locks" you out and you must wait (lockout time is set at company level) before you try again. If the system displays the *SIGNON NAME NOT AUTHORIZED* message, contact your S1 security administrator.

### 9.8.3.1 Archive/Retrieval Menu (VJARMENU) Screen

**Overview:** The VJARMENU screen provides a menu of the formats to be used in conjunction with archive and retrieval processes. By returning to this screen after performing tasks on another format, the user can select a format appropriate for the next work activity.

**NOTE –** The VJARFNSD and VJARTLST formats are not listed on the VJARMENU screen; they are accessed by means of PF key functions from other formats in the course of normal work flow.

```
ARCHIVE/RETRIEVAL MENU SCREEN (VJARMENU)          /FOR --------
                                                  02/26/93 11:24:04

     (ENTER ANY CHARACTER TO SELECT)

   - VJARDSPA   UPDATE/VIEW ARCHIVE SELECTION

   - VJARDSPR   UPDATE/VIEW RETRIEVAL SELECTION

   - VJARLSTA   VIEW ARCHIVE SELECTION LIST SCREEN

   - VJARLSTR   VIEW RETRIEVAL SELECTION LIST SCREEN

   - VJARCNTL   UPDATE/VIEW ARCHIVE CONTROL CARDS
```

**Figure 9–11.** Archive/Retrieval Menu (VJARMENU) Screen

**Table 9–8.** How to Display a Blank VJARMENU Screen

| From This Screen: | Do This: |
|---|---|
| From a cleared terminal screen: | Type **/FOR VJARMENU** and press the **<ENTER>** key. |
| From a NSDB screen other than VJARMENU: | Type **VJARMENU** in the /FOR field and press the **<ENTER>** key. |

## Selecting a Format via the VJARMENU Screen

There are two methods of making a format selection from the Archive/Retrieval Menu screen. You can choose either of these options:

1. Tab to the hidden field at the left of the particular menu option you wish to access.
2. Type a character (any character) in this field.
3. Press the **<ENTER>** key.

*OR*

1. In the */FOR* field, type the name of the format you wish to access.
2. Press the **<ENTER>** key.

The requested format will be returned.

**Table 9–10.** How to Display a Blank VJARCNTL Screen

| From This Screen: | Do This: |
|---|---|
| From a cleared terminal screen: | Type **/FOR VJARCNTL** and press the **<ENTER>** key. |
| From a VJARCNTL screen on which data is currently being displayed: | Press the REFRESH **<PF8>** function key. |
| From the VJARMENU screen: | Select the "UPDATE/VIEW ARCHIVE CONTROL CARDS" option and press the **<ENTER>** key. |
| From a NSDB screen other than VJARCNTL: | Type **VJARCNTL** in the /FOR field and press the **<ENTER>** key. This automatic JUMP/FIND feature is designed to save time when the user needs to move to a different screen. NSDB will attempt to populate the new VJARCNTL screen with information associated with the original screen. A blank screen will be returned if an association cannot be made. |

### Displaying a Control Card Image with Aging Criteria for an Archive Run

The unique combination of the archive name and run date and time serve to identify the particular control card image with aging criteria for a given archive request. To display the aging criteria for an archive selection request, follow these instructions:

1. In the *ARCHIVE NAME:* field, type the desired archive name.
2. In the *RUN DATE/TIME:* field, enter the date and time for which the archive run is scheduled.

   If the run date and time are not entered, the system will default this field to zeros, which indicates that the request has not been run.

   **NOTE –** Requests which *have not* been run *can* be updated. Conversely, requests which *have* been run *cannot* be updated.
3. Press the FIND **<PF1>** function key.

The information displayed on the VJARCNTL that returns will be used to control the next archive run if the control card image is associated with an outstanding selection request.

### Displaying the Next Page of Control Card Images

If there are more than 14 control card images for an archive request, you can display the additional pages by pressing the FORWARD **<PF2>** function key.

## Displaying the Previous Page of Control Card Images

The BACK <**PF3**> function key can be pressed to return to a previously displayed page of control card images when the request results span multiple pages.

## Adding a Control Card Image to a Archive Selection Request

For NSDB, a control card image is used to define the age factor for archiving selected data from the Audit database, the Service Order Image database, and the Line Record database.

To add (enter) a control card image containing aging criteria to the VJARCNTL screen, follow these steps:

1. In the *ARCHIVE NAME:* field, type the desired archive name(unless it has been prepopulated with the desired data).
2. The *RUN DATE/TIME:* field should be left blank.

   The system will default this field to zeros, which is an indication that the request has not been run.
3. In the first three characters of the first control card image line, type the number representing the age factor in terms of days. For instance, an aging criteria of 45 days would be entered as **045**, as shown in the screen sample for the VJARCNTL screen.

   The age factor is typically set to fall within a range of 30 to 90 days, depending on the daily volume of records. A minimum factor of 30 days has been established for normal work operations. If a value of less than 30 days is entered, it will be automatically converted by the system to a value of 30 days. The 30 day limitation can be overridden for the NSDB product by appending "OVERRIDE AGE DEFAULT" to the desired age factor (e.g., **015 OVERRIDE AGE DEFAULT**).

   Additional notes can be typed in the *CONTROL CARD IMAGES* section, but they will not be processed by the system. The notes will, however, be stored and displayed upon request of the VJARCNTL screen.
4. Press the <**PF4**> ADD function key.

Aging criteria is now defined for your archive request.

## Updating Control Card Images

Individual or multiple control card images can be changed, inserted, or deleted from the VJARCNTL screen. For NSDB, it is expected that the user will most frequently use the update option to change age criteria.

To *change* a control card image,

1. Display the VJARCNTL format associated with the desired archive name (refer to the section "Displaying a Control Card Image with Aging Criteria for an Archive Run").
2. In the *SEL* field to the left of the target control card image, type a **C** (Change) and overtype the aging criteria with the desired information or use the space bar to eliminate unnecessary narrative.
3. Press the UPDATE **<PF5>** function key.

To *insert* a control card image after an existing image,

1. Display the VJARCNTL format associated with the desired archive name (refer to the section "Displaying a Control Card Image with Aging Criteria for an Archive Run").
2. In the *SEL* field to the left of the existing control card image, type an **I** (Insert).
3. Press the UPDATE **<PF5>** function key.

The screen will then display open lines below the selected image for the user to enter new image information. Only one control card image is defined for the the Archive/Retrieval feature associated with NSDB, but additional notes can be typed in the *CONTROL CARD IMAGES* section. The notes will not, however, be processed by the system. They will be stored and displayed upon request of the VJARCNTL screen.

To *delete* a control card image,

1. Display the VJARCNTL format associated with the desired archive name (refer to the section "Displaying a Control Card Image with Aging Criteria for an Archive Run").
2. In the *SEL* field to the left of the control card image to be deleted, type a **D** (Delete).
3. Press the UPDATE **<PF5>** function key.

The entire line comprising the control card image will be deleted.

## Field Descriptions

### /FOR

Full Name: Format

Definition: Identifies the name of the screen to which the user wishes to jump.

Data Content: to 8 A/N

Example: **VJARDSPA**

### ARCHIVE NAME:

Full Name: Archive Name

Definition: The name of the archive run into which the run control cards will be input.

Data Content: 8 A/N

Valid Entry:

| Valid Entries: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

### RUN DATE/TIME:

Full Name: Run Date and Time

Definition: The date and time when the archive selection request was processed. If the run date and time are left blank, the system will default this field to zeros, which indicates that the request has not been run.

Data Content: 8 A/N – 8 A/N

Example: **02/26/93 07:30:00** (A completed selection request)
**00/00/00 00:00:00** (An outstanding selection request)

### CREATE DATE/TIME:

| | |
|---|---|
| Full Name: | Create Date and Time |
| Definition: | The date and time when the archive selection request was created. |
| Data Content: | 8 A/N – 8 A/N (protected field) |
| Example: | **02/26/93 07:00:00** |

### SEL

| | |
|---|---|
| Full Name: | Selection |
| Definition: | Indicates the selection made for processing the control card image. |
| Data Content: | 1A |
| Valid Entries: | **I** – Insert<br>**C** – Change<br>**D** – Delete |

### CONTROL CARD IMAGES

| | |
|---|---|
| Full Name: | Control Card Images |
| Definition: | Each 72-character line in this section comprises a control card image which is used to enter run control information in the format predetermined by application-specific program logic. For NSDB, a valid control card image consists of three numeric characters at the beginning of the first line in the control card image section. This represents the number of days to be used by the system as aging criteria for archiving data from select databases.<br>Additional notes can be typed in the *CONTROL CARD IMAGES* section, but they will not be processed by the system. The notes will, however, be stored and displayed upon a FIND of the VJARCNTL screen. |
| Data Content: | 3 N |
| Example: | **030** (30 days) |

**NOTE —** The age factor is typically set to fall within a range of 30 to 90 days, depending on the daily volume of records. A minimum factor of 30 days has been established for normal work operations. If a value of less than 30 days is entered, it will be automatically converted by the system to a value of 30 days. The 30 day limitation can be overridden for the NSDB product by appending "OVERRIDE AGE DEFAULT" to the desired age factor (e.g., **015 OVERRIDE AGE DEFAULT**).

### 9.8.3.3 Archive Selection Display (VJARDSPA) Screen

**Overview:** The VJARDSPA screen is used to identify any special criteria needed to select records for archive.

This should be done only if archiving by age criteria (via the VJARCNTL screen) is not adequate for your needs.

```
 PAGE: 001        ARCHIVE SELECTION DISPLAY (VJARDSPA)          /FOR --------
                                                              02/26/93 09:59:55
   ARCHIVE NAME : --------                                    TAPE BACKUP: -
   RUN DATE/TIME: -------- --------    CREATE DATE/TIME:

C   FIELD      OP          VALUE
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------
- ------------ -- ------------------------------------------------------------

  PFKEYS: 1=FIND 2=FWD 3=BACK 4=ADD 5=UPDATE 7=APPEND 8=REFRESH 9=LIST
          10=DELETE 11=MENU (UPDATE LINE COMMANDS: I=INSERT D=DELETE C=CHANGE)
```

**Figure 9–13.** Archive Selection Display (VJARDSPA) Screen

**Table 9–11.** PF Keys and Commands to Use on the VJARDSPA Screen

| Press This Key: | For This Function: | To Do This: |
|---|---|---|
| <PF1> | *FIND* | Display an existing archive selection request. |
| <PF2> | *FORWARD* | Page forward to the next page of the archive selection request. |
| <PF3> | *BACK* | Page back to the previous page of the archive selection request. |
| <PF4> | *ADD* | Create the first archive selection request for a given archive name or jump to the VJARFNSD screen to display and select valid Comparison Field Names. |
| <PF5> | *UPDATE* | Update the archive selection request that is currently displayed [Line Commands: C (Change), I (Insert), and D (Delete)]. |
| <PF6> | | N/A |
| <PF7> | *APPEND* | Jump to the VJARFNSD format to select and append valid Comparison Field Names. |
| <PF8> | *REFRESH* | Clear all data from the screen. |
| <PF9> | *LIST* | Jump to the Archive Selection List (VJARLSTA) screen to display the selection requests for the current archive name. |
| <PF10> | *DELETE* | Delete the selection criteria for an outstanding archive selection request that is currently displayed. |
| <PF11> | *MENU* | Jump to the Archive/Retrieval Menu (VJARMENU) screen. |

**Table 9–12.** How to Display a Blank VJARDSPA Screen

| From This Screen: | Do This: |
|---|---|
| From a cleared terminal screen: | Type **/FOR VJARDSPA** and press the **<ENTER>** key. |
| From a VJARDSPA screen on which data is currently being displayed: | Press the REFRESH **<PF8>** function key. |
| From the VJARMENU screen: | Select the "UPDATE/VIEW ARCHIVE SELECTION" option and press the **<ENTER>** key. |
| From a NSDB screen other than VJARDSPA: | Type **VJARDSPA** in the /FOR field and press the **<ENTER>** key. This automatic JUMP/FIND feature is designed to save time when the user needs to move to a different screen. NSDB will attempt to populate the new VJARDSPA screen with information associated with the original screen. A blank screen will be returned if an association cannot be made. |

## Displaying an Existing Archive Selection Request

The unique combination of the archive name and run date and time serve to identify a particular archive selection request. To display an archive selection request, follow these instructions:

1. In the *ARCHIVE NAME:* field, type the desired archive name.
2. In the *RUN DATE/TIME:* field, enter the date and time for which the archive run is scheduled.

   If the run date and time are not entered, the system will default this field to zeros, which indicates that the request has not been run.

   > **NOTE** – Requests which *have not* been run *can* be updated or deleted. Conversely, requests which *have* been run *cannot* be updated or deleted.

3. Press the FIND **<PF1>** function key.

## Displaying the Next Page of an Archive Selection Request

If the archive selection request is too large to be contained on a single page, you can display the additional pages by pressing the **<PF2>** FORWARD function key.

## Displaying the Previous Page of an Archive Selection Request

The **<PF3>** BACK function key can be pressed to return to a previously displayed page of an archive selection request when the request spans multiple pages.

## Adding an Archive Selection Request

To create the first archive selection request for an archive name, follow these instructions:

1. In the *ARCHIVE NAME* field, type the appropriate archive name.
2. In the field designated as *FIELD*, type the field name to be evaluated during the archiving process.

   *OR*

   If you are unfamiliar with the fields available for defining the parameter values or if you need to select several of these fields for the archive selection request, you can press the ADD **<PF4>** function key after entering information in the *ARCHIVE NAME:* field before typing any selection criteria. You will be jumped to the Fieldname Selection Display (VJARFNSD) format which contains valid key and non-key fields that can be used to define this special selection criteria. After selecting fields on the VJARFNSD screen, you can transfer them to the *FIELD* locations on the VJARDSPA screen and proceed with these instructions. [Refer to the section "Fieldname Selection Display (VJARFNSD) Screen".]
3. In the *OP* field, type the logical operator to be associated with the *FIELD* value.
4. In the *VALUE* field, type the literal value to be compared against the *FIELD* value [by means of the logical operator (*OP*)].
5. 

   **NOTE –** It is not necessary to clear the remaining unused fields before adding the request since any field without an associated operator and value will not be processed as a part of the archive selection request.

   Press the ADD **<PF4>** function key.

The archive selection request is generated. The *RUN DATE/TIME:* will default to zeros, indicating that the request has not yet been run.

If the selection criteria needed to define the archive selection request are too numerous to be contained on a single page, you can perform an ADD **<PF4>** function for the first page and APPEND **<PF7>** functions for additional pages.

Creating an archive selection request by means of special selection criteria is not the formidable task it may first seem. The sample VJARDSPA screen below is followed by a detailed explanation which should help if you decide to use the special selection criteria option.

```
PAGE:            ARCHIVE SELECTION DISPLAY (VJARDSPA)          /FOR
                                                          02/26/93 13:02:51
  ARCHIVE NAME : VJDBALR1                                 TAPE BACKUP:
  RUN DATE/TIME:                        CREATE DATE/TIME:

C   FIELD       OP          VALUE
  CAC           SS 1,1
                EQ S
                OR
                EQ N









  PFKEYS: 1=FIND 2=FWD 3=BACK 4=ADD 5=UPDATE 7=APPEND 8=REFRESH 9=LIST
          10=DELETE 11=MENU (UPDATE LINE COMMANDS: I=INSERT D=DELETE C=CHANGE)
```

**Figure 9–14.** Sample Archive Selection Display (VJARDSPA) Screen

You must have specific knowledge of field attributes in order to construct usable archive selection requests via special selection criteria. The Archive Selection Display format shown above provides an example of an archive selection request based on special selection criteria. The first point worthy of note is that the sequence of selection criteria lines is a most important consideration due to the hierarchical effects of logical operators "AND" and "OR". An AND operator is implied between lines of selection criteria unless an OR separates them. (For those unfamiliar with logical operators, an AND creates an *intersection* of criteria, thus limiting the selection of records; the OR operator creates a *union* of criteria, thus enlarging the pool of eligible records.)

In this sample, the requester asks that the system consider only those records with CAC values beginning with the letters "S" or "N". The purpose of this request is to archive completed records in the Line Record database for both designed and nondesigned circuits.

The first line of selection criteria includes an SS (SubString) operator that works in conjunction with the next three lines of CAC detail. For *FIELD*, *OP* and *VALUE* entries of CAC, SS and 1,1 respectively, the system will use a substring that starts with the first

character of the CAC value and extends one character from that point. The substring line in this sample sets the stage for lines two and four which specify that only CAC codes starting with "S" or "N" are to be included in this archive request. Notice that lines two and four are linked with the OR operator. The OR operator overrides the implied AND operator that is usually in control. In contrast to the limiting effect of the AND operator, the OR operator serves to broaden the pool of eligible records. The OR operator on the third line provides a means to include several values for a single CAC entry.

It should be apparent by now that one must have a strong knowledge of field attributes in order to construct a meaningful archive request via special selection criteria.

## Updating an Archive Selection Request

Only requests which are outstanding can be updated. To update an archive selection request, follow these instructions:

1. Display the target archive selection request with a FIND by following the instructions listed in the section "Displaying an Existing Archive Selection Request".
2. Perform the needed updating function (change, insert or delete). Only one *type* of function can be entered at a time during a given UPDATE request.
   - CHANGE – In the *C* field at the beginning of the targeted selection criteria line, type a **C** (Change).

     The system will not allow changes to lines that contain a value in the *FIELD* column. The system *does* allow multiple changes to lines that do not contain a value in the *FIELD* column.
   - INSERT – In the *C* field at the beginning of the targeted selection criteria line, type an **I** (Insert). Only one insert function can be performed at a time.
     — If the **I** is entered on a line that is followed by a line with a *FIELD* value, you will be able to insert new field names after step 3.
     — If the **I** is entered on a line that is followed by a line without a *FIELD* value, you will not be allowed to insert new field names. (In other words, the system will not permit the addition of a new field in the midst of existing field selection criteria.)

     **NOTE –** You may enter a maximum of 14 lines of selection criteria for a field value.
   - DELETE – In the *C* field at the beginning of the selection criteria line to be deleted, type a **D** (Delete).

     Multiple deletes are permitted. If the **D** is entered on a line with a *FIELD* value, the system will delete all associated selection criteria lines.

3. Press the UPDATE **<PF5>** function key.

   The system will give a response appropriate to the type of function you are requesting.

   - If you requested a Change (**C**), a screen will be displayed highlighting the data that can be revised. You can then overtype the Comparison Field Name, Operator, or Comparison Value with the desired information, or use the space bar to eliminate unnecessary narrative.
   - If you requested an Insert (**I**), a screen will be displayed that shows open lines beneath the selected image for the entry of new archive selection information. You can proceed by typing new data on these lines.
   - If you requested a Delete (**D**), a screen will be displayed requesting that you press the UPDATE **<PF5>** function key again for verification purposes.

4. Press the UPDATE **<PF5>** function key a second time.

The archive selection request will be modified according to your instructions. If additional changes are needed, you must redisplay the request by using the FIND **<PF1>** function key.

Only one page of the archive selection request can be updated at a time. If you need to access additional pages in order to complete the update process, you can use the FORWARD **<PF2>** function key to continue to the next page.

## Appending to an Archive Selection Request

To append selection criteria to an outstanding archive selection request, follow these steps:

1. Display the target archive selection request with a FIND by following the instructions listed in the section "Displaying an Existing Archive Selection Request".
2. Press the APPEND **<PF7>** function key.

   You will be jumped to the Fieldname Selection Display (VJARFNSD) format which contains valid key and non-key fields that can be used to define this special selection criteria. After selecting fields on the VJARFNSD screen, you can transfer them to the *FIELD* locations on the VJARDSPA screen and proceed with these instructions. [Refer to the section "Fieldname Selection Display (VJARFNSD) Screen".]

3. In the *OP* field, type the logical operator to be associated with each *FIELD* value.

4. In the *VALUE* field, type the literal value to be compared against each *FIELD* value [by means of the logical operator (*OP*)].
5. Press the APPEND **<PF7>** function key a second time.

The selection request is modified according to your instructions.

## Deleting All Selection Criteria for an Archive Selection Request

To delete all selection criteria for an existing archive selection request, follow these steps:

1. Display the target archive selection request by following the instructions listed in the section "Displaying an Existing Archive Selection Request".
2. Press the DELETE **<PF10>** function key.

   The system responds with a request to press the DELETE **<PF10>** function key a second time for verification purposes.
3. Press the DELETE **<PF10>** function key again.

The selection criteria for the archive selection request is deleted from the Archive Selection database (although control card criteria remains).

> **NOTE –** To delete individual criteria from an archive selection request, the user can apply the UPDATE command, as discussed in the section "Updating an Archive Selection Request".

## Field Descriptions

### /FOR

Full Name: Format

Definition: Identifies the name of the screen to which the user wishes to jump.

Data Content: to 8 A/N

Example: **VJARLSTR**

### ARCHIVE NAME:

Full Name: Archive Name

Definition: Archive Name The name of the archive process used to generate the archive file. This name is used to match against the TTS tables A/R AFD and A/R RFD in order to obtain the appropriate field names for display or validation.

Data Content: 8 A/N

Valid Entry:

| Valid Entries: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

### TAPE BACKUP:

Full Name: Tape Backup

Definition: At archive time, the archived data can be stored on a primary tape or on both a primary tape and a backup tape, depending on the option selected for this flag.

Data Content: 1 A

Valid Entries: **Y** [Yes] (Primary *and* backup tapes)
**N** [No] (Primary tape only)

### RUN DATE/TIME:

| | |
|---|---|
| Full Name: | Run Date and Time |
| Definition: | The date and time when the archive selection request was processed. If the run date and time are left blank, the system will default this field to zeros, which indicates that the request has not been run. |
| Data Content: | 8 A/N – 8 A/N |
| Example: | **02/26/93 07:30:00** (A completed selection request)<br>**00/00/00 00:00:00** (An outstanding selection request) |

### CREATE DATE/TIME:

| | |
|---|---|
| Full Name: | Create Date and Time |
| Definition: | The date and time when the archive selection request was created. |
| Data Content: | 8 A/N – 8 A/N (protected field) |
| Example: | **02/26/93 07:00:00** |

The following four fields collectively form one Selection Criteria entry. An "Archive Selection Request" is a compilation of multiple Selection Criteria entries. Up to 14 Selection Criteria entries can be displayed on a single VJARDSPA format.

### C

| | |
|---|---|
| Full Name: | Line Command |
| Definition: | Indicates the selection made for updating the archive selection request. |
| Data Content: | 1 A |
| Valid Entries: | **I** – Insert<br>**C** – Change<br>**D** – Delete |

### FIELD

| | |
|---|---|
| Full Name: | Comparison Field Name |
| Definition: | The field in a searchable database that is to be scanned for a particular value. It may be a key or non-key field, as defined in the section "TTS Tables Supporting Archive/Retrieve Functionality". |
| Data Content: | to 12 A/N |
| Example: | **CAC** |

## OP

Full Name: Comparison Operator

Definition: The logical operation to be performed between the Comparison Field Name (*FIELD*) and the Comparison Value (*VALUE*).

**NOTE —** A logical "AND" operator is implied between lines of special selection criteria unless the OR operator is specified. The logical AND operator creates an *intersection* of consecutive *VALUE*s; it has the effect of *limiting* the data to be archived because data must meet each listed criteria in order to be considered for archive.

Data Content: 2 A

Valid Entries:

| Valid Entry: | Meaning: |
|---|---|
| EQ | Equal to |
| GE | Greater than or equal to |
| GT | Greater than |
| LE | Less than or equal to |
| LT | Less than |
| NE | Not equal to |
| OR | Logical "OR"; this creates a *union* of the values listed above and below it; the OR operator has the effect of *increasing* the data to be archived to include any additional OR'd values. (The OR operator overrides the logical "AND" operator that is implied for each line of special selection criteria.) |
| SS | Substring (The substring operator is used to define the portion of the *FIELD* entry to be used in the selection criteria.) |

**NOTE —** Experienced users may elect to employ standard PL/I logical operators instead of the alphabetic operators shown in the table above.

## VALUE

| | |
|---|---|
| Full Name: | Comparison Value |
| Definition: | The constant value to be compared against the Comparison Field Name (*FIELD*) using the Comparison Operator (*OP*).<br><br>The *VALUE* entry associated with the SS operator defines the starting position and the length of the substring. For example, if the *FIELD*, *OP*, and *VALUE* entries are listed as CAC, SS and 1,1 respectively, then for all following *OP* and *VALUE* entries for CAC, a substring of CAC that begins with the first character and has a length of one character will be used for processing. |
| Data Content: | 61 A/N |
| Example: | **SMN4MDA3** |

### 9.8.3.4 Fieldname Selection Display (VJARFNSD) Screen

**Overview:** The Fieldname Selection Display (VJARFNSD) format permits display and selection of the fields that can be used to define special selection criteria for either archive requests or retrieval requests.

```
 PAGE: 001        FIELDNAME SELECTION DISPLAY (VJARFNSD)        /FOR
                                                              02/26/93 10:11:10
   ARCHIVE NAME : VJDBALR1  REPORT ID:                        TAPE BACKUP: N
   RUN DATE/TIME: 00/00/00 00:00:00    CREATE DATE/TIME: 02/26/93 12:58:59

S   FIELD      S   FIELD      S   FIELD      S   FIELD      S   FIELD

  CID            WFA_CMPLDATE   TRK            CAC            ACNA
  OT             ORDNO          CUST           MCN            BTN
  LL             AA             ACO            ACTL           ADL
  ADSR           AGENT          AGENT_TN       AR             AU
  AUTHOR         AVN            BA             BL             BN
  BRG            BRT            BSD            BS1            BS2
  CACT           CCF#           CCNA           CCO            CCON
  CKLEND         CKT_SRC        CLS            CLSV           CLT
  CO             CORS           CS             CSN            CSR
  CTEL           CTG            CTID           CTL_CTID       CTX
  CUS            DAA            DBPT           DD             DND
  DNP            DPTN           DROP           DSGCON         DSNA
  DTEL           DTN            ECT            EMCT           ES
  ESL            ESV            FF_ID          FF_ID2         FF_ID3
 PFKEYS:  2=FWD 3=BACK 4=PROCESS 9=LIST 11=MENU
 *** KEY FIELDS HIGHLIGHTED ***
ARR043I: VALID KEY FIELD NAMES APPEAR ABOVE. PAGE FORWARD FOR MORE FIELDS.
```

**Figure 9–15.** Sample Fieldname Selection Display (VJARFNSD) Screen

**Table 9–13.** PF Keys and Commands to Use on the VJARFNSD Screen

| Press This Key: | For This Function: | To Do This: |
|---|---|---|
| <PF1> | | N/A |
| <PF2> | *FORWARD* | Page forward to the next page of fields that can be used to define special archive or retrieval selection criteria. |
| <PF3> | *BACK* | Page back to the previous page of fields that can be used to define special archive or retrieval selection criteria. |
| <PF4> | *PROCESS* | Carry the marked fields that are to be used for special archive or retrieval selection criteria to the respective VJARDSPA or VJARDSPR format for further processing. |
| <PF5> | | N/A |
| <PF6> | | N/A |
| <PF7> | | N/A |
| <PF8> | | N/A |
| <PF9> | *LIST* | Jump to the Archive Selection List (VJARLSTA) screen or Retrieval Selection List (VJARLSTR) screen, as appropriate. |
| <PF10> | | N/A |
| <PF11> | *MENU* | Jump to the Archive/Retrieval Menu (VJARMENU) screen. |

## Selecting Fields for Special Archive Criteria

To select the fields you wish to use for archive or retrieval criteria, follow these instructions:

1. The VJARFNSD screen is accessed by means of the VJARDSPA or VJARDSPR screens, as described below.
   - From the VJARDSPA screen, press the ADD **<PF4>** function key after entering information in the *ARCHIVE NAME:* field.
   - From the VJARDSPR screen, press the ADD **<PF4>** function key after entering information in both the *ARCHIVE NAME:* and *REPORT ID:* fields.
   - From either the VJARDSPA or VJARDSPR screen, press the APPEND **<PF7>** function key after performing a FIND **<PF1>** function.

     You will be jumped to the corresponding VJARFNSD format.
2. In the *S* (Selection Indicator) field(s) to the left of the listed field names, type any character to select the field(s) you wish to use in archive or retrieval selection criteria.

   **NOTE –** Key fields are highlighted on the VJARFNSD screen. You must include at least *one* key field in your selection of fields for retrieval via special criteria.
3. To carry the marked fields from the displayed VJARFNSD screen back to the VJARDSPA or VJARDSPR screen, press the PROCESS **<PF4>** function key.

The VJARDSPA or VJARDSPR screen that returns will display the selected fields in the column under *FIELD*.

## Field Descriptions

### /FOR

Full Name: Format

Definition: Identifies the name of the screen to which the user wishes to jump.

Data Content: to 8 A/N

Example: **VJARDSPA**

### ARCHIVE NAME:

Full Name: Archive Name

Definition: The name of the archive process used to generate the archive file. This name is used to match against the TTS tables A/R AFD and A/R RFD in order to obtain the appropriate field names for display or validation.

Data Content: 8 A/N

Valid Entry:

| Valid Entries: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

### REPORT ID:

Full Name: Report Identification

Definition: The name designated to represent the unique retrieval selection request associated with the *ARCHIVE NAME.*

Data Content: to 8 A/N

Example: **CIRCUITS**

### TAPE BACKUP:

| | |
|---|---|
| Full Name: | Tape Backup |
| Definition: | At archive time, the archived data can be stored on a primary tape or on both a primary tape and a backup tape, depending on the option selected for this flag. |
| Data Content: | 1 A (protected field) |
| Valid Entries: | **Y** [Yes] (Primary *and* backup tapes)<br>**N** [No] (Primary tape only) |

### RUN DATE/TIME:

| | |
|---|---|
| Full Name: | Run Date and Time |
| Definition: | The date and time when the archive or retrieval selection request was processed. |
| Data Content: | 8 A/N – 8 A/N (protected field) |
| Example: | **02/26/93 07:30:00** (A completed selection request)<br>**00/00/00 00:00:00** (An outstanding selection request) |

### CREATE DATE/TIME:

| | |
|---|---|
| Full Name: | Create Date and Time |
| Definition: | The date and time when the archive or retrieval selection request was created. |
| Data Content: | 8 A/N – 8 A/N (protected field) |
| Example: | **02/26/93 07:00:00** |

### S

| | |
|---|---|
| Full Name: | Selection Indicator |
| Definition: | This field permits the user to select any field displayed on the VJARFNSD screen for inclusion in the special archive or retrieval selection criteria. |
| Data Content: | 1 A/N (any character) |
| Examples: | **X**, * |

## FIELD

| | |
|---|---|
| Full Name: | Comparison Field Name |
| Definition: | A column containing the fields that can be used to define special archive or retrieval selection criteria. Each field is included in the searchable database(s) that is to be scanned for a particular value. They may be key or non-key fields, as defined in the section "TTS Tables Supporting Archive/Retrieve Functionality". |
| Data Content: | to 12 A/N (protected field) |
| Examples: | **CAC** |

### 9.8.3.5 Archive Selection List (VJARLSTA) Screen

Overview: The Archive Selection List screen (VJARLSTA) is used to list current and previous archive selections, run dates and times, and creation dates and times. The user may select one of the requests from this list for further processing.

```
                ARCHIVE SELECTION LIST (VJARLSTA)            /FOR --------
                                                          02/26/93 10:05:49
          ARCHIVE NAME: --------
          FROM RUN DATE: --------     TO RUN DATE: --------     (MM/DD/YY)

   SEL RUN DATE     RUN TIME      USERID     CREATE DATE  CREATE TIME

     -
     -
     -
     -
     -
     -
     -
     -
     -
     -
     -
     -
     -

PFKEYS: 1=FIND 2=FWD 3=BACK 4=VIEW 6=CCINFO 8=REFRESH 10=DELETE 11=MENU
```

**Figure 9–16.** Archive Selection List (VJARLSTA) Screen

**Table 9–14.** PF Keys and Commands to Use on the VJARLSTA Screen

| Press This Key: | For This Function: | To Do This: |
|---|---|---|
| <PF1> | *FIND* | List the archive selection requests for the archive name and run time range entered by the user. |
| <PF2> | *FORWARD* | Page forward to the next page of archive selection requests. |
| <PF3> | *BACK* | Page back to the previous page of archive selection requests. |
| <PF4> | *VIEW* | Allows the user to jump to the corresponding Archive Selection Display (VJARDSPA) screen to view and update archive selection criteria. |
| <PF5> | | N/A |
| <PF6> | *CCINFO* | Allows the user to jump to the corresponding Archive Control Card (VJARCNTL) screen to view and update run-specific parameters. |
| <PF7> | | N/A |
| <PF8> | *REFRESH* | Clear all data from the screen. |
| <PF9> | | N/A |
| <PF10> | *DELETE* | Allows the user to delete archive selection criteria and control card information (aging criteria). |
| <PF11> | *MENU* | Jump to the Archive/Retrieval Menu (VJARMENU) screen. |

**Table 9–15.** How to Display a Blank VJARLSTA Screen

| From This Screen: | Do This: |
|---|---|
| From a cleared terminal screen: | Type **/FOR VJARLSTA** and press the **<ENTER>** key. |
| From a VJARLSTA screen on which data is currently being displayed: | Press the REFRESH **<PF8>** function key. |
| From the VJARMENU screen: | Select the "VIEW ARCHIVE SELECTION LIST SCREEN" option and press the **<ENTER>** key. |
| From the VJARCNTL, VJARDSPA, and VJARFNSD screens: | Press the **<PF9>** LIST function key. NSDB will attempt to populate the new VJARLSTA screen with data from the original screen. A blank screen will be returned if an association cannot be made. |
| From a NSDB screen other than VJARLSTA: | Type **VJARLSTA** in the /FOR field and press the **<ENTER>** key. This automatic JUMP/FIND feature is designed to save time when the user needs to move to a different screen. NSDB will attempt to populate the new VJARLSTA screen with information associated with the original screen. A blank screen will be returned if an association cannot be made. |

## Listing Archive Selection Requests

To list archive selection requests for a given archive name, follow these instructions:

1. In the *ARCHIVE NAME:* field, type the name of the archive process that is used to generate the archive file.
2. In the *FROM RUN DATE:* field, type the earliest date of the range in which the desired archive selection requests were processed.

   **NOTE** – If the *FROM RUN DATE:* and *TO RUN DATE:* fields are left blank, then all archive selection requests for the designated archive name will be displayed.
3. In the *TO RUN DATE:* field, type the latest date of the range in which the archive selection requests were processed.
4. Press the FIND **<PF1>** function key.

The archive selection requests that fall within the specified criteria will be displayed on the screen.

## Displaying the Next Page of Archive Selection Requests

If there are more than 14 archive selection requests meeting the list criteria, you can display the additional pages by pressing the FORWARD **<PF2>** function key.

## Displaying the Previous Page of Archive Selection Requests

The BACK **<PF3>** function key can be pressed to return to a previously displayed page of archive selection requests when the record spans multiple pages.

## Jumping to the VJARDSPA Screen for Viewing and Updating Archive Selection Requests (Special Selection Criteria)

To jump to the Archive Selection Display (VJARDSPA) screen from the VJARLSTA screen for the purpose of viewing and/or updating special selection criteria for an archive request, follow these steps:

1. In the *SEL* field to the left of the target archive request, type any single character.
2. Press the VIEW **<PF4>** function key.

An Archive Selection Display (VJARDSPA) screen will return which includes data corresponding to the request selected on the VJARLSTA format. You can update the archive selection request via the VJARDSPA format if you desire [refer to the section "Archive Selection Display (VJARDSPA) Screen" for details].

> **NOTE** – Only outstanding selection requests can be updated. These are indicated by zeros in the *RUN DATE* and *RUN TIME* fields.

## Jumping to the VJARCNTL Screen for Viewing and Updating Archive Selection Requests (Aging Criteria)

To jump to the Archive Control Card (VJARCNTL) screen from the VJARLSTA screen for the purpose of viewing and/or updating aging criteria for an archive request, follow these steps:

1. In the *SEL* field to the left of the target archive request, type any single character.
2. Press the CCINFO **<PF6>** function key.

An Archive Control Card (VJARCNTL) screen will return which includes data corresponding to the request selected on the VJARLSTA format. You can update the archive selection request via the VJARCNTL format if you desire [refer to the section "Archive Control Card (VJARCNTL) Screen" for details].

> **NOTE** – Only outstanding selection requests can be updated. These are indicated by zeros in the *RUN DATE* and *RUN TIME* fields.

## Deleting Control Card Information and Special Selection Criteria for an Archive Selection Request

To delete control card information *and* special selection criteria for an outstanding archive selection request, follow these steps:

> **NOTE** – To delete only the control card information, refer to the section "Archive Control Card (VJARCNTL) Screen". To delete only the special selection criteria, refer to the section "Archive Selection Display (VJARDSPA) Screen".

1. In the *SEL* field to the left of the target archive request, type any single character.
2. Press the DELETE **<PF10>** function key.

Control card information and special selection criteria for the target archive selection request is deleted.

## Field Descriptions

### /FOR

Full Name: Format

Definition: Identifies the name of the screen to which the user wishes to jump.

Data Content: to 8 A/N

Example: **VJARLSTR**

### ARCHIVE NAME:

Full Name: Archive Name

Definition: The name of the archive process used to generate the archive file. This name is used to match against the TTS tables A/R AFD and A/R RFD in order to obtain the appropriate field names for display or validation.

Data Content: 8 A/N

Valid Entry:

| Valid Entries: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

### FROM RUN DATE:

Full Name: From Run Date

Definition: Earliest date of the range during which the archive selection requests were processed.

Data Content: 8 A/N

Example: **02/26/93** (Sample for a completed selection request)
**00/00/00** (Outstanding selection request)

### TO RUN DATE:

| | |
|---|---|
| Full Name: | To Run Date |
| Definition: | Latest date of the range during which the archive selection requests were processed. |
| Data Content: | 8 A/N |
| Example: | **02/26/93** (Sample for a completed selection request)<br>**00:00:00** (Outstanding selection request) |

### SEL

| | |
|---|---|
| Full Name: | Selection |
| Definition: | This field permits the user to jump to another format to revise processing options for an archive selection request. |
| Data Content: | 1 A/N |
| Examples: | **X**, **U** (any single character) |

### RUN DATE

| | |
|---|---|
| Full Name: | Run Date |
| Definition: | Date when the archive selection request was processed. |
| Data Content: | 8 A/N (protected field) |
| Example: | **02/26/93** (Sample for a completed selection request)<br>**00/00/00** (Outstanding selection request) |

### RUN TIME

| | |
|---|---|
| Full Name: | Run Time |
| Definition: | Time when the archive selection request was processed. |
| Data Content: | 8 A/N (protected field) |
| Example: | **10:00:00** (Sample for a completed selection request)<br>**00:00:00** (Outstanding selection request) |

### USERID

| | |
|---|---|
| Full Name: | User Identification |
| Definition: | The unique designation that identifies the person who entered the archive selection request. |
| Data Content: | to 8 A (protected field) |
| Examples: | **TSOUSER** |

### CREATE DATE

| | |
|---|---|
| Full Name: | Create Date |
| Definition: | Date when the archive selection request was created. |
| Data Content: | 8 A/N (protected field) |
| Example: | **02/26/93** |

### CREATE TIME

| | |
|---|---|
| Full Name: | Create Time |
| Definition: | Time when the archive selection request was created. |
| Data Content: | 8 A/N (protected field) |
| Example: | **12:23:43** |

### 9.8.3.6 Retrieval Selection Display (VJARDSPR) Screen

**Overview:** The VJARDSPR screen is used to view, create, and update retrieval selection requests.

```
PAGE:            RETRIEVAL SELECTION DISPLAY (VJARDSPR)        /FOR --------
                                                              02/26/93 10:06:33
  ARCHIVE NAME : --------- REPORT ID: --------               TAPE BACKUP: -
  RUN DATE/TIME: -------- --------     CREATE DATE/TIME:

C   FIELD      OP          VALUE

- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
- ---------------- -- ----------------------------------------------------------
  PFKEYS: 1=FIND 2=FWD 3=BACK 4=ADD 5=UPDATE 7=APPEND 8=REFRESH 9=LIST
          10=DELETE 11=MENU (UPDATE LINE COMMANDS: I=INSERT D=DELETE C=CHANGE)
```

**Figure 9–17.** Retrieval Selection Display (VJARDSPR) Screen

**Table 9–16.** PF Keys and Commands to Use on the VJARDSPR Screen

| Press This Key: | For This Function: | To Do This: |
|---|---|---|
| <PF1> | *FIND* | Display an existing retrieval selection request. |
| <PF2> | *FORWARD* | Page forward to the next page of the retrieval selection criteria. |
| <PF3> | *BACK* | Page back to the previous page of the retrieval selection criteria. |
| <PF4> | *ADD* | Create a retrieval selection request for a given archive name or jump to the VJARFNSD screen to display and select valid Comparison Field Names. |
| <PF5> | *UPDATE* | Update the retrieval selection request that is currently displayed [Line Commands: C (Change), I (Insert), and D (Delete)]. |
| <PF6> | | N/A |
| <PF7> | *APPEND* | Jump to the VJARFNSD format to display and append valid Comparison Field Names. |
| <PF8> | *REFRESH* | Clear all data from the screen. |
| <PF9> | *LIST* | Jump to the Retrieval Selection List (VJARLSTR) screen to display selection requests for the current archive name. |
| <PF10> | *DELETE* | Delete the selection criteria for an outstanding retrieval selection request that is currently displayed. |
| <PF11> | *MENU* | Jump to the Archive/Retrieval Menu (VJARMENU) screen. |

Table 9–17. How to Display a Blank VJARDSPR Screen

| From This Screen: | Do This: |
|---|---|
| From a cleared terminal screen: | Type /FOR VJARDSPR and press the **<ENTER>** key. |
| From a VJARDSPR screen on which data is currently being displayed: | Press the REFRESH **<PF8>** function key. |
| From the VJARMENU screen: | Select the "UPDATE/VIEW RETRIEVAL SELECTION" option and press the **<ENTER>** key. |
| From a NSDB screen other than VJARDSPR: | Type **VJARDSPR** in the /FOR field and press the **<ENTER>** key. This automatic JUMP/FIND feature is designed to save time when the user needs to move to a different screen. NSDB will attempt to populate the new VJARDSPR screen with information associated with the original screen. A blank screen will be returned if an association cannot be made. |

### Displaying an Existing Retrieval Selection Request

The unique combination of the archive name, report ID, and run date and time serve to identify a particular retrieval selection request. To display a retrieval selection request, follow these instructions:

1. In the *ARCHIVE NAME:* field, type the desired archive name.
2. In the *REPORT ID:* field, type the report identification name since there may be multiple retrieval requests for the archive name you have selected.
3. In the *RUN DATE/TIME:* field, enter the date and time for which the archive run is scheduled.

   If the run date and time are not entered, the system will default this field to zeros, which indicates that the request has not been run.

   **NOTE** – Requests which *have not* been run *can* be updated or deleted. Conversely, requests which *have* been run *cannot* be updated or deleted.
4. Press the FIND **<PF1>** function key.

### Displaying the Next Page of a Retrieval Selection Request

If the retrieval selection request is too large to be contained on a single page, you can display the additional pages by pressing the **<PF2>** FORWARD function key.

## Displaying the Previous Page of a Retrieval Selection Request

The <PF3> BACK function key can be pressed to return to a previously displayed page of a retrieval selection request when the request spans multiple pages.

## Adding a Retrieval Selection Request

To create a retrieval selection request for an archive name/report ID, follow these instructions:

1. In the *ARCHIVE NAME* field, type the appropriate archive name.
2. In the *REPORT ID:* field, type the report identification name you wish to associate with this unique request for the archive name.
3. In the field designated as *FIELD*, type the field name to be evaluated during the retrieval process.

   OR

   If you are unfamiliar with the fields available for defining the parameter values or if you need to select several of these fields for the retrieval selection request, you can press the ADD <PF4> function key (after entering information in the *ARCHIVE NAME:* and *REPORT ID:* fields) before typing any selection criteria. You will be jumped to the Fieldname Selection Display (VJARFNSD) format which contains the valid key and non-key fields which can be used to define this special selection criteria. After selecting the fields (including at least one key field) on the VJARFNSD screen, you can transfer them to the *FIELD* locations on the VJARDSPR screen and proceed with these instructions. [Refer to the section "Fieldname Selection Display (VJARFNSD) Screen".]
4. In the *OP* field, type the logical operator to be associated with the *FIELD* value.
5. In the *VALUE* field, type the literal value to be compared against the *FIELD* value [by means of the logical operator (*OP*)].
6. 

   **NOTE –** It is not necessary to clear the remaining unused fields before adding the request since any field without an associated operator and value will not be processed as a part of the retrieval selection request.

   Press the ADD <PF4> function key.

The retrieval selection request is generated. The *RUN DATE/TIME:* will default to zeros, indicating that the request has not yet been run.

If the selection criteria needed to define the retrieval selection request are too numerous to be contained on a single page, you can perform an ADD **<PF4>** function for the first page and APPEND **<PF7>** functions for additional pages.

Creating a retrieval selection request by means of special selection criteria is not the formidable task it may first seem. The sample VJARDSPR screen below is followed by a detailed explanation which should help if you decide to use the special selection criteria option.

```
PAGE:            RETRIEVAL SELECTION DISPLAY (VJARDSPR)        /FOR
                                                          02/26/93 13:02:51
  ARCHIVE NAME : VJDBALR1  REPORT ID: CIRCUITS            TAPE BACKUP: N
  RUN DATE/TIME:                      CREATE DATE/TIME:

C   FIELD      OP          VALUE
 CAC           SS 1,1
               EQ S
               OR
               EQ N









 PFKEYS: 1=FIND 2=FWD 3=BACK 4=ADD 5=UPDATE 7=APPEND 8=REFRESH 9=LIST
         10=DELETE 11=MENU (UPDATE LINE COMMANDS: I=INSERT D=DELETE C=CHANGE)
```

**Figure 9–18.** Sample Retrieval Selection Display (VJARDSPR) Screen

You must have specific knowledge of field attributes in order to construct usable retrieval selection requests via special selection criteria. The Retrieval Selection Display format shown above provides an example of a retrieval selection request based on special selection criteria. The first point worthy of note is that the sequence of selection criteria lines is a most important consideration due to the hierarchical effects of logical operators "AND" and "OR". An AND operator is implied between lines of selection criteria unless an OR separates them. (For those unfamiliar with logical operators, an AND creates an *intersection* of criteria, thus limiting the selection of records; the OR operator creates a *union* of criteria, thus enlarging the pool of eligible records.)

In this sample, the requester asks that the system consider only those records with CAC values beginning with the letters "S" or "N". The purpose of this request is to retrieve completed records in the Line Record database for both designed and nondesigned circuits.

The first line of selection criteria includes an SS (SubString) operator that works in conjunction with the next three lines of CAC detail. For *FIELD*, *OP* and *VALUE* entries of CAC, SS and 1,1 respectively, the system will use a substring that starts with the first character of the CAC value and extends one character from that point. The substring line in this sample sets the stage for lines two and four which specify that only CAC codes starting with "S" or "N" are to be included in this retrieval request. Notice that lines two and four are linked with the OR operator. The OR operator overrides the implied AND operator that is usually in control. In contrast to the limiting effect of the AND operator, the OR operator serves to broaden the pool of eligible records. The OR operator on the third line provides a means to include several values for a single CAC entry.

It should be apparent by now that one must have a strong knowledge of field attributes in order to construct a meaningful retrieval request via special selection criteria.

## Updating a Retrieval Selection Request

Only requests which are outstanding can be updated. To update a retrieval selection request, follow these instructions:

1. Display the target retrieval selection request with a FIND by following the instructions listed in the section "Displaying an Existing Retrieval Selection Request".
2. Perform the needed updating function (change, insert or delete). Only one *type* of function can be entered at a time during a given UPDATE request.
   - CHANGE – In the *C* field at the beginning of the targeted selection criteria line, type a **C** (Change).

     The system will not allow changes to lines that contain a value in the *FIELD* column. The system *does* allow multiple changes to lines that do not contain a value in the *FIELD* column.
   - INSERT – In the *C* field at the beginning of the targeted selection criteria line, type an **I** (Insert). Only one insert function can be performed at a time.
     — If the **I** is entered on a line that is followed by a line with a *FIELD* value, you will be able to insert new field names after step 3.
     — If the **I** is entered on a line that is followed by a line without a *FIELD* value, you will not be allowed to insert new field names. (In other words, the system will not permit the addition of a new field in the midst of existing field selection criteria.)

       **NOTE** – You may enter a maximum of 14 lines of selection criteria for a field value.

- DELETE – In the *C* field at the beginning of the selection criteria line to be deleted, type a **D** (Delete).

  Multiple deletes are permitted. If the **D** is entered on a line with a *FIELD* value, the system will delete all associated selection criteria lines.

3. Press the UPDATE **<PF5>** function key.

   The system will give a response appropriate to the type of function you are requesting.

   - If you requested a Change (**C**), a screen will be displayed highlighting the data that can be revised. You can then overtype the Comparison Field Name, Operator, or Comparison Value with the desired information, or use the space bar to eliminate unnecessary narrative.
   - If you requested an Insert (**I**), a screen will be displayed that shows open lines beneath the selected image for the entry of new retrieval selection information. You can proceed by typing new data on these lines.
   - If you requested a Delete (**D**), a screen will be displayed requesting that you press the UPDATE **<PF5>** function key again for verification purposes.

4. Press the UPDATE **<PF5>** function key a second time.

The retrieval selection request will be modified according to your instructions. Tapelist information will be deleted since it is no longer valid. If additional changes are needed, you must redisplay the request by using the FIND **<PF1>** function key.

Only one page of the retrieval selection request can be updated at a time. If you need to access additional pages in order to complete the update process, you can use the FORWARD <PF2> function key to continue to the next page.

## Appending to a Retrieval Selection Request

To append selection criteria to an outstanding retrieval selection request, follow these steps:

1. Display the target retrieval selection request with a FIND by following the instructions listed in the section "Displaying an Existing Retrieval Selection Request".
2. Press the APPEND **<PF7>** function key.

   You will be jumped to the Fieldname Selection Display (VJARFNSD) format which contains valid key and non-key fields that can be used to define this special selection criteria. After selecting fields (including at least one key field) on the VJARFNSD screen, you can transfer them to the *FIELD* locations on the VJARDSPR screen and proceed with these instructions. [Refer to the section "Fieldname Selection Display (VJARFNSD) Screen".]

3. In the *OP* field, type the logical operator to be associated with each *FIELD* value.
4. In the *VALUE* field, type the literal value to be compared against each *FIELD* value [by means of the logical operator (*OP*)].
5. Press the APPEND **<PF7>** function key a second time.

The selection request is modified according to your instructions. Tapelist information will be deleted since it is no longer valid.

## Deleting All Selection Criteria for a Retrieval Selection Request

To delete all selection criteria for an existing retrieval selection request, follow these steps:

1. Display the target retrieval selection request by following the instructions listed in the section "Displaying an Existing Retrieval Selection Request".
2. Press the DELETE **<PF10>** function key.

   The system responds with a request to press the DELETE **<PF10>** function key a second time for verification purposes.
3. Press the DELETE **<PF10>** function key again.

The selection criteria for the retrieval selection request is deleted from the Retrieval Selection database. Tapelist information will also be deleted since it is no longer valid.

> **NOTE –** To delete individual criteria from a retrieval selection request, the user can apply the UPDATE command, as discussed in the section "Updating a Retrieval Selection Request".

## Field Descriptions

### /FOR

Full Name: Format

Definition: Identifies the name of the screen to which the user wishes to jump.

Data Content: to 8 A/N

Example: **VJARTLST**

### ARCHIVE NAME:

Full Name: Archive Name

Definition: The name of the archive process used to generate the archive file. This name is used to match against the TTS tables A/R AFD and A/R RFD in order to obtain the appropriate field names for display or validation.

Data Content: 8 A/N

Valid Entry:

| Valid Entries: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

### REPORT ID:

Full Name: Report Identification

Definition: The name designated to represent the unique retrieval selection request associated with the *ARCHIVE NAME.*

Data Content: to 8 A/N

Example: **CIRCUITS**

### TAPE BACKUP:

| | |
|---|---|
| Full Name: | Tape Backup |
| Definition: | At archive time, the archived data can be stored on a primary tape or on both a primary tape and a backup tape, depending on the option selected for this flag. |
| Data Content: | 1 A |
| Valid Entries: | **Y** [Yes] (Primary *and* backup tapes)<br>**N** [No] (Primary tape only) |

### RUN DATE/TIME:

| | |
|---|---|
| Full Name: | Run Date and Time |
| Definition: | The date and time when the the retrieval selection request was processed. If the run date and time are left blank, the system will default this field to zeros, which indicates that the request has not been run. |
| Data Content: | 8 A/N – 8 A/N |
| Example: | **02/26/93 07:30:00** (A completed selection request)<br>**00/00/00 00:00:00** (An outstanding selection request) |

### CREATE DATE/TIME:

| | |
|---|---|
| Full Name: | Create Date and Time |
| Definition: | The date and time when the the retrieval selection request was created. |
| Data Content: | 8 A/N – 8 A/N (protected field) |
| Example: | **02/26/93 07:00:00** |

The following four fields collectively form one Selection Criteria entry. A "Retrieval Selection Request" is a compilation of multiple Selection Criteria entries. Up to 14 Selection Criteria entries can be displayed on a single VJARDSPR format.

## C

| | |
|---|---|
| Full Name: | Line Command |
| Definition: | Indicates the selection made for updating the retrieval selection request. |
| Data Content: | 1A |
| Valid Entries: | **I** – Insert<br>**C** – Change<br>**D** – Delete |

## FIELD

| | |
|---|---|
| Full Name: | Comparison Field Name |
| Definition: | The field in a searchable database that is to be scanned for a particular value. It may be a key or non-key field, as defined in the section "TTS Tables Supporting Archive/Retrieve Functionality". |
| Data Content: | to 12 A/N |
| Examples: | **CAC** |

## OP

Full Name: Comparison Operator

Definition: The logical operation to be performed between the Comparison Field Name (*FIELD*) and the Comparison Value (*VALUE*).

**NOTE** — A logical "AND" operator is implied between lines of special selection criteria unless the OR operator is specified. The logical AND operator creates an *intersection* of consecutive *VALUEs*; it has the effect of *limiting* the data to be retrieved because data must meet each listed criteria in order to be considered for retrieval.

Data Content: 2 A

Valid Entries:

| Valid Entry: | Meaning: |
|---|---|
| EQ | Equal to |
| GE | Greater than or equal to |
| GT | Greater than |
| LE | Less than or equal to |
| LT | Less than |
| NE | Not equal to |
| OR | Logical "OR"; this creates a *union* of the values listed above and below it; the OR operator has the effect of *increasing* the data to be retrieved to include any additional OR'd values. (The OR operator overrides the logical "AND" operator that is implied for each line of special selection criteria.) |
| SS | Substring (The substring operator is used to define the portion of the *FIELD* entry to be used in the selection criteria.) |

**NOTE** — Experienced users may elect to employ standard PL/I logical operators instead of the alphabetic operators shown in the table above.

## VALUE

| | |
|---|---|
| Full Name: | Comparison Value |
| Definition: | The constant value to be compared against the Comparison Field Name (*FIELD*) using the Comparison Operator *OP*).<br><br>The *VALUE* entry associated with the SS operator defines the starting position and the length of the substring. For example, if the *FIELD*, *OP*, and *VALUE* entries are listed as CAC, SS and 1,1 respectively, then for all following *OP* and *VALUE* entries for CAC, a substring of CAC that begins with the first character and has a length of one character will be used for processing. |
| Data Content: | 61 A/N |
| Example: | **SMN4MDA3** |

### 9.8.3.7 Retrieval Selection List (VJARLSTR) Screen

**Overview:** The Retrieval Selection List screen (VJARLSTR) is used to list current and previous retrieval selections, run dates and times, creation dates and times, and the number of records to be retrieved during the retrieval process. The user may select one of the requests from this list for further processing.

**NOTE –** If you wish to view the number of records to be retrieved for an *outstanding* retrieval selection request, you must first execute a tape identification BMP run.

```
                RETRIEVAL SELECTION LIST (VJARLSTR)          /FOR --------
                                                          02/26/93 10:08:52
             ARCHIVE NAME: --------
            FROM RUN DATE: --------     TO RUN DATE: --------    (MM/DD/YY)

 SEL REPORT-ID  RUN DATE   RUN TIME    USERID   CREATE DATE CREATE TIME   RECS

  -
  -
  -
  -
  -
  -
  -
  -
  -
  -
  -
  -
  -
  -

 PFKEYS: 1=FIND 2=FWD 3=BACK 4=VIEW 6=TAPELIST 8=REFRESH 10=DELETE 11=MENU
```

**Figure 9–19.** Retrieval Selection List (VJARLSTR) Screen

**Table 9–18.** PF Keys and Commands to Use on the VJARLSTR Screen

| Press This Key: | For This Function: | To Do This: |
|---|---|---|
| <PF1> | *FIND* | List the retrieval selection requests for the archive name and run time range entered by the user. |
| <PF2> | *FORWARD* | Page forward to the next page of retrieval selection requests. |
| <PF3> | *BACK* | Page back to the previous page of retrieval selection requests. |
| <PF4> | *VIEW* | Allows the user to jump to the corresponding Retrieval Selection Display (VJARDSPR) screen to view and update retrieval selection criteria. |
| <PF5> | | N/A |
| <PF6> | *TAPELIST* | Allows the user to jump to the corresponding Archive/Retrieval Tapelist (VJARTLST) screen to view tape volume-serial numbers for a retrieval selection request. |
| <PF7> | | N/A |
| <PF8> | *REFRESH* | Clear all data from the screen. |
| <PF9> | | N/A |
| <PF10> | *DELETE* | Allows the user to delete retrieval selection criteria and tapelist information. |
| <PF11> | *MENU* | Jump to the Archive/Retrieval Menu (VJARMENU) screen. |

**Table 9–19.** How to Display a Blank VJARLSTR Screen

| From This Screen: | Do This: |
|---|---|
| From a cleared terminal screen: | Type **/FOR VJARLSTR** and press the **<ENTER>** key. |
| From a VJARLSTR screen on which data is currently being displayed: | Press the REFRESH **<PF8>** function key. |
| From the VJARMENU screen: | Select the "VIEW RETRIEVAL SELECTION LIST SCREEN" option and press the **<ENTER>** key. |
| From the VJARDSPR, VJARFNSD, and VJARTLST screens: | Press the **<PF9>** LIST function key. NSDB will attempt to populate the new VJARLSTR screen with data from the original screen. A blank screen will be returned if an association cannot be made. |
| From a NSDB screen other than VJARLSTR: | Type **VJARLSTR** in the /FOR field and press the **<ENTER>** key. This automatic JUMP/FIND feature is designed to save time when the user needs to move to a different screen. NSDB will attempt to populate the new VJARLSTR screen with information associated with the original screen. A blank screen will be returned if an association cannot be made. |

## Listing Retrieval Selection Requests

To list archive selection requests for a given archive name, follow these instructions:

1. In the *ARCHIVE NAME:* field, type the name of the archive process that was used to generate the archive file.
2. In the *FROM RUN DATE:* field, type the earliest date of the range in which the desired retrieval selection requests were processed.

   **NOTE** – If the *FROM RUN DATE:* and *TO RUN DATE:* fields are left blank, then all retrieval selection requests for the designated archive name will be displayed.

3. In the *TO RUN DATE:* field, type the latest date of the range in which the retrieval selection requests were processed.
4. Press the FIND **<PF1>** function key.

The retrieval selection requests that fall within the specified criteria will be displayed on the screen.

## Displaying the Next Page of Retrieval Selection Requests

If there are more than 14 retrieval selection requests meeting the list criteria, you can display the additional pages by pressing the FORWARD **<PF2>** function key.

## Displaying the Previous Page of Retrieval Selection Requests

The BACK **<PF3>** function key can be pressed to return to a previously displayed of retrieval selection requests when the record spans multiple pages.

## Jumping to the VJARDSPR Screen for Viewing and Updating Retrieval Selection Requests (Special Selection Criteria)

To jump to the Retrieval Selection Display (VJARDSPR) screen from the VJARLSTR screen for the purpose of viewing and/or updating special selection criteria for a retrieval request, follow these steps:

1. In the *SEL* field to the left of the target retrieval request, type any single character.
2. Press the VIEW **<PF4>** function key.

A Retrieval Selection Display (VJARDSPR) screen will be displayed which includes data corresponding to the request selected on the VJARLSTR format. You can update the retrieval selection request via the VJARDSPR format if you desire [refer to the section "Retrieval Selection Display (VJARDSPR) Screen" for details].

> **NOTE** – Only outstanding selection requests can be updated. These are indicated by zeros in the *RUN DATE* and *RUN TIME* fields.

## Jumping to the VJARTLST Screen for Viewing Tape Volume-Serial Numbers

To jump to the Archive/Retrieval Tapelist (VJARTLST) from the VJARLSTR screen for the purpose of viewing the tape volume-serial numbers for a retrieval selection request, follow these steps:

1. In the *SEL* field to the left of the target retrieval request, type any single character.
2. Press the TAPELIST **<PF6>** function key.

An Archive/Retrieval Tapelist (VJARTLST) screen will be displayed which includes data corresponding to the request selected on the VJARLSTR format.

## Deleting Special Selection Criteria and Tapelist Information for a Retrieval Selection Request

To delete special selection criteria and tapelist information for an outstanding retrieval selection request, follow these steps:

1. In the *SEL* field to the left of the target retrieval request, type any single character.
2. Press the DELETE **<PF10>** function key.

Special selection criteria and tapelist information for the specified selection request is deleted.

## Field Descriptions

### /FOR

Full Name: Format

Definition: Identifies the name of the screen to which the user wishes to jump.

Data Content: to 8 A/N

Example: **VJARLSTA**

### ARCHIVE NAME:

Full Name: Archive Name

Definition: The name of the archive process used to generate the archive file. This name is used to match against the TTS tables A/R AFD and A/R RFD in order to obtain the appropriate field names for display or validation.

Data Content: 8 A/N

Valid Entry:

| Valid Entries: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

### FROM RUN DATE:

Full Name: From Run Date

Definition: Earliest date of the range during which the retrieval selection requests were processed.

Data Content: 8 A/N

Example: **02/26/93** (Sample for a completed selection request)
**00/00/00** (Outstanding selection request)

### TO RUN DATE:

| | |
|---|---|
| Full Name: | To Run Date |
| Definition: | Latest date of the range during which the retrieval selection requests were processed. |
| Data Content: | 8 A/N |
| Example: | **02/26/93** (Sample for a completed selection request)<br>**00:00:00** (Outstanding selection request) |

### SEL

| | |
|---|---|
| Full Name: | Selection |
| Definition: | This field permits the user to jump to another format to revise processing options for a retrieval selection request. |
| Data Content: | 1 A/N |
| Examples: | **X, U** (any single character) |

### REPORT-ID

| | |
|---|---|
| Full Name: | Report Identification |
| Definition: | The name of the report to which the selected data will be written. |
| Data Content: | to 8 A/N (protected field) |
| Example: | **CIRCUITS** |

### RUN DATE

| | |
|---|---|
| Full Name: | Run Date |
| Definition: | Date when the retrieval selection request was processed. |
| Data Content: | 8 A/N (protected field) |
| Example: | **02/26/93** (Sample for a completed selection request)<br>**00/00/00** (Outstanding selection request) |

### RUN TIME

| | |
|---|---|
| Full Name: | Run Time |
| Definition: | Time when the retrieval selection request was processed. |
| Data Content: | 8 A/N (protected field) |
| Example: | **10:00:00** (Sample for a completed selection request)<br>**00:00:00** (Outstanding selection request) |

### USERID

| | |
|---|---|
| Full Name: | User Identification |
| Definition: | The unique designation that identifies the person who entered the retrieval selection request. |
| Data Content: | to 8 A (protected field) |
| Examples: | **TSOUSER** |

### CREATE DATE

| | |
|---|---|
| Full Name: | Create Date |
| Definition: | Date when the retrieval selection request was created. |
| Data Content: | 8 A/N (protected field) |
| Example: | **02/26/93** |

### CREATE TIME

| | |
|---|---|
| Full Name: | Create Time |
| Definition: | Time when the retrieval selection request was created. |
| Data Content: | 8 A/N (protected field) |
| Example: | **12:23:43** |

### RECS

| | |
|---|---|
| Full Name: | Records |
| Definition: | The number of records for a retrieval selection request that is retrieved by the retrieval process. |

**NOTE —** If you wish to view the number of records to be retrieved for an *outstanding* retrieval selection request, you must first execute a tape identification BMP run.

| | |
|---|---|
| Data Content: | 6 N (protected field) |
| Examples: | **3, 21** |

### 9.8.3.8 Archive/Retrieval Tapelist (VJARTLST) Screen

**Overview:** The Archive/Retrieval Tapelist screen (VJARTLST) is used to display tape volume-serial numbers for the tapes on which data was written by the archive process for the specified archive name and report ID. This information can be viewed only after the execution of a tape identification run and before the designated retrieval run has been performed. By reviewing retrieval selection requests before they are processed, the user can verify that the needed tapes are available. The user can also review any request that has an unusually high or low volume.

**NOTE –** If you wish to view the tape volume-serial numbers and the number of records to be retrieved for an *outstanding* retrieval selection request, you must first execute a tape identification BMP run.

```
                ARCHIVE/RETRIEVAL TAPELIST (VJARTLST)        /FOR --------
                                                          02/26/93 10:09:32
 ARCHIVE NAME :                      REPORT ID:
 TID DATE/TIME:                      CREATE DATE/TIME:

                        TAPE VOLUME-SERIAL NUMBERS
--------------------------------------------------------------------------------




 PFKEYS: 2=FWD 3=BACK 9=LIST 11=MENU
```

**Figure 9–20.** Archive/Retrieval Tapelist (VJARTLST) Screen

**Table 9–20.** PF Keys and Commands to Use on the VJARTLST Screen

| Press This Key: | For This Function: | To Do This: |
|---|---|---|
| <PF1> | | N/A |
| <PF2> | *FORWARD* | Page forward to the next page of tapelist information. |
| <PF3> | *BACK* | Page back to the previous page of tapelist information. |
| <PF4> | | N/A |
| <PF5> | | N/A |
| <PF6> | | N/A |
| <PF7> | | N/A |
| <PF8> | | N/A |
| <PF9> | *LIST* | Jump to the Retrieval Selection List (VJARLSTR) screen. |
| <PF10> | | N/A |
| <PF11> | *MENU* | Jump to the Archive/Retrieval Menu (VJARMENU) screen. |

## Accessing the Archive/Retrieval Tapelist (VJARTLST) Screen

*The only way to access the VJARTLST format is via the VJARLSTR screen.* After a retrieval selection request has been selected on the VJARLSTR screen, you can press the TAPELIST **<PF6>** function key to jump to the associated VJARTLST screen.

> **NOTE –** The VJARTLST format is *not* accessible from other online screens via the */FOR* field.

### Displaying the Next Page of Tapelist Information

If the tapelist information exceeds a single page, you can display additional pages by pressing the FORWARD **<PF2>** function key.

## Displaying the Previous Page of Tapelist Information

The BACK **<PF3>** function key can be pressed to return to a previously displayed page of tapelist information when the record spans multiple pages.

## Field Descriptions

### /FOR

Full Name: Format

Definition: Identifies the name of the screen to which the user wishes to jump.

Data Content: to 8 A/N

Example: **VJARLSTA**

### ARCHIVE NAME:

Full Name: Archive Name

Definition: The name of the archive process used to generate the archive file. This name is used to match against the TTS tables A/R AFD and A/R RFD in order to obtain the appropriate field names for display or validation.

Data Content: 8 A/N (protected field)

Valid Entry:

| Valid Entries: | For this database: |
|---|---|
| **VJDBAAU1** | Audit database |
| **VJDBADN1** | Designed circuit details databases – *(In–effect Circuit/Carrier, Circuit Index, and Message)* |
| **VJDBAIM1** | Service Order Image database |
| **VJDBALR1** | Line Record database |

### REPORT ID:

Full Name: Report Identification

Definition: The name of the report to which the selected data will be written.

Data Content: to 8 A/N (protected field)

Example: **CIRCUITS**

### TID DATE/TIME:

| | |
|---|---|
| Full Name: | Tape Identification Date and Time |
| Definition: | Date and time when the last tape identification process was run for this retrieval request. |
| Data Content: | 8 A/N – 8 A/N (protected field) |
| Example: | **02/26/93 10:33:43** |

### CREATE DATE/TIME:

| | |
|---|---|
| Full Name: | Create Date and Time |
| Definition: | Date and time when the archive selection request was created. |
| Data Content: | 8 A/N – 8 A/N (protected field) |
| Example: | **02/26/93 08:15:44** |

### TAPE VOLUME-SERIAL NUMBERS

| | |
|---|---|
| Full Name: | Tape Volume-Serial Numbers |
| Definition: | The volume-serial number of a tape on which data was written in the BMP run for the archive name. |

**NOTE** — If you wish to view the tape volume-serial numbers and the number of records to be retrieved for an *outstanding* retrieval selection request, you must first execute a tape identification BMP run.

| | |
|---|---|
| Data Content: | 6 A/N (protected field) |
| Example: | **X30475** |

### 9.8.4 Archiving by Age – A Sample Procedure

If you want to archive by age criteria only, follow the sample procedure below:

1. Access the Archive Control Card (VJARCNTL) screen via the main menu(VJARMENU), or by typing **/FOR VJARCNTL** and pressing the **<ENTER>** key.
2. On the VJARCNTL screen that returns, type an *ARCHIVE NAME* value for NSDB; for this example, let's use **VJDBALR1** for the Line Record database.
3. Use the tab function to place the cursor below the dotted line and one tab movement to the right of the *SEL* field. The cursor should be positioned five vertical spaces under the letter "*H*" in the *ARCHIVE NAME* field.
4. Type the age (in days) of the records to be archived; use right justification for this three-character value. Typing **090**, for example, means that every completed record that is 90 days old or older will be archived.
5. Press the ADD **<PF4>** function key.
   - If an outstanding Control Card already exists for the Archive Name, the system will ask you to perform a FIND **<PF1>** function. This is to protect the existing Control Card from accidental changes. If you still want to change the age value that appears on the existing Control Card Image, type the letter **C** in the *SEL* field, and then overtype the existing age value with the new age value. Press the UPDATE **<PF5>** function key to update the Control Card Image. To verify the change, press the FIND **<PF1>** function key. The new value will be reflected on the VJARCNTL screen.
   - If no outstanding Control Card exists for the Archive Name used, the ADD in step 5 will be successful.

### 9.8.5 Archiving by Age and Field Criteria – A Sample Procedure

If you want to archive both by aging criteria and field-specific criteria, follow the sample procedure below. (The first part of this procedure is similar to the procedure described above in which age is the only factor considered.)

1. Access the Archive Control Card (VJARCNTL) screen via the main menu(VJARMENU), or by typing **/FOR VJARCNTL** and pressing the **<ENTER>** key.
2. On the VJARCNTL screen that returns, type an *ARCHIVE NAME* value for NSDB; for this example, let's use **VJDBALR1** for the Line Record database.
3. Use the tab function to place the cursor below the dotted line and one tab movement to the right of the *SEL* field. The cursor should be positioned five vertical spaces under the letter "*H*" in the *ARCHIVE NAME* field.
4. Type in the age (in days) of the records to be archived; use right justification for this three-character value. Typing **090**, for example, means every completed record that is 90 days old or older will be archived.
5. Press the ADD **<PF4>** function key.
   - If an outstanding Control Card already exists for the Archive Name, the system will ask you to perform a FIND **<PF1>** function. This is to protect the existing Control Card from accidental changes. If you still want to change the age value that appears on the existing Control Card Image, type the letter **C** in the *SEL* field, and then overtype the existing age value with the new age value. Press the UPDATE **<PF5>** function key to update the Control Card Image. To verify the change, press the FIND **<PF1>** function key. The new value will be reflected on the VJARCNTL screen.
   - If no outstanding Control Card exists for the Archive Name used, the ADD in step 5 will be successful.
6. From the Archive Control Card (VJARCNTL) screen, use the LIST **<PF9>** function key to jump to the Archive Selection List (VJARLSTA) screen. You can also jump to VJARLSTA screen by using the */FOR* field.
7. From the populated VJARLSTA screen, tab down to the first *SEL* field. The *RUN DATE* and *RUN TIME* fields to the right of the cursor should show zeros. Type any character in the *SEL* field and press the VIEW **<PF4>** function key. This should cause you to jump to the Archive Selection Display (VJARDSPA) screen.
8. In this example, we will build a field-specific selection request that will archive all completed records in the Line Record database meeting the aging criteria of 90 days (in this example) for completed designed circuits [Circuit Access Code (CAC) of "S"]. To start, use the tab key to position the cursor *below* and *between* the fields *C* and *FIELD*. This is where the value for *FIELD* is entered. Type **CAC** in the field designated *FIELD*.

9. Tab to the *OP* field. In the *OP* (Operator) field, type **SS** to indicate that a substring will be used.
10. Notice the cursor jumped one tab space after the entry "SS" in the *OP* field. Your cursor is actually at the beginning of the *VALUE* field. Type **1,1** in the *VALUE* field.
11. Tab twice to position the cursor on the second line of selection criteria, and type **EQ** in the *OP* field to indicate that the CAC code convention defined in the line above must "equal" the value to follow. It is not necessary to re-enter **CAC** as a *FIELD* value; the system will assume the *FIELD* value to be the same as the previous line if none is listed on the current line.
12. In the *VALUE* field on the second line, type an **S** to specify designed circuits.
13. Press the ADD **<PF4>** function key.

The following message will be displayed on line 24 of the VJARDSPA screen: "*ARR033I: RECORD ADDED SUCCESSFULLY*".

The system will archive records in the Line Record database for designed circuits that were completed at least 90 days ago.

### 9.8.6 Retrieval by Field Criteria – A Sample Procedure

If you want to construct a retrieval request, you can use the sample procedure below to guide you.

> **NOTE** – A retrieval selection request must be constructed by establishing field-specific selection criteria. You will need to incorporate at least one key field (the fields highlighted on the VJARFNSD screen) in your retrieval request. The system will process your request most efficiently if the request uses as many key fields as possible to narrow the scope of retrievable records.

In this example, we will build a field-specific selection request that will retrieve all archived records in the Line Record database for designed circuits. Refer to the following instructions:

1. Access a blank Retrieval Selection Display (VJARDSPR)screen via the main menu(VJARMENU), or by typing **/FOR VJARDSPR** and pressing the **<ENTER>** key. (The VJARDSPR format controls the retrieval and tape identification BMP runs.)
2. On the VJARDSPR screen that returns, type an *ARCHIVE NAME* value for NSDB; for this example, let's use **VJDBALR1** for the Line Record database.
3. In the *REPORT ID* field, type the name (up to 8 A/N characters) you have chosen to represent this unique retrieval request.
4. Use the tab key to position the cursor *below* and *between* the fields *C* and *FIELD*. This is where the value for *FIELD* is entered. Type **CAC** in the field designated *FIELD*.
5. Tab to the *OP* field. In the *OP* (Operator) field, type **SS** to indicate that a substring will be used.
6. Notice the cursor jumped one tab space after the entry "SS" in the *OP* field. Your cursor is actually at the beginning of the *VALUE* field. Type **1,1** in the *VALUE* field.
7. Tab twice to position the cursor on the second line of selection criteria, and type **EQ** in the *OP* field to indicate that the CAC code convention defined in the line above must "equal" the value to follow. It is not necessary to re-enter **CAC** as a *FIELD* value; the system will assume the *FIELD* value to be the same as the previous line if none is listed on the current line.
8. In the *VALUE* field on the second line, type an **S** to specify designed circuits.
9. Press the ADD **<PF4>** function key.

# Appendixes
# CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# 10. Appendixes

## 10.1 Appendix A – Online Validation Message Directory

This section describes the messages that NSDB can pass as the result of a batch/BMP database validation run. These messages appear on the following reports that are generated by the validation run:

- Bulk Edit For NSDB Databases – DDS Summary Report
- Bulk Edit For NSDB Databases – DS1 Summary Report
- Bulk Edit For NSDB Databases – CLFI Summary Report

### 10.1.1 DDS Validations

Full Name: ODB250E CIRCUIT CONTAINS TOO MANY MJUs

Data Source: This message is returned when more than two MJUs are detected on a circuit. NSDB and ITS compare MJU HECIG codes on EM lines (inventoried) and E* or M lines (non-inventoried) with those in the ODB HECIG TABLE, to recognize MJUs.

Full Name: ODB251E MULTI-POINT CIRCUIT MISSING MJU RECORD(S)

Data Source: This message is passed when, on a multi-point circuit (segment information different than 1, 2, A, B), no E*, M, EM, or EU line assignment information can be detected.

Full Name: ODB252E MJU OUT OF TRANSMISSION SEQUENCE

Data Source: On a multi-point circuit, NSDB checks the first and last lines of the design for EM, E*, or M lines (which reflect the MJU assignments). If one is found, the design is considered a master/remote leg configuration. If two are found, the design is considered to be a cascading section. These are the conditions that would cause this message to be passed:

— For a master or remote leg, there is no valid termination (determined by NI or S lines) on the opposite end of the design.

— For a cascading section circuit, the MJU assignments are not the first and last lines on the design.

Full Name: ODB253E MJU LOCATION, RELAY RACK, OR UNIT NUM ARE BLANK

Data Source: This message is passed when, on a multi-point circuit, NSDB detects a valid MJU HECIG code but cannot find any assignment information (i.e., Location, Relay Rack, Unit Num).

Full Name: ODB254E CIRCUIT HAS NOT BEEN PROPERLY TERMINATED

Data Source: In a multipoint circuit, the MJU assignment must be in the first or last lines of the circuit details. A valid termination must also be in the first or last position, but opposite to the MJU. This message is returned when an MJU is in the first position but a valid termination is not in the last (or visa versa).

Full Name: ODB255E BOTH MJUS ARE ASSIGNED TO PORT ZERO

Data Source: NSDB first determines if a circuit is multi-point by checking for MJU assignments. If one MJU exists, it is considered a master or a remote leg. If two MJUs exist, the circuit is considered a cascading section. On a cascading section of a multi-point circuit, NSDB checks the first and last lines of the design for EM, E*, or M lines (reflecting the MJU assignments). NSDB then checks the port assignments of the two bridges. This message is passed if both port assignments are equal to zero. This decision is based on the assumption that the master leg, input, of an MJU should always be zero.

Full Name: ODB256E UNABLE TO PARSE SUBDIV-MISC FIELD FOR PORT NUMBER

Data Source: NSDB passes this message if the MJU port number is not a valid format as stated in the NSDB ODB CO OPTIONS TTS table.

Full Name: ODB257E CIRCUIT MISSING OCU RECORDS(S)

Data Source: This message is passed if any of the following circumstances occur:

- *In a multi-point (master/remote leg) circuits* – the number of OCUs using "P" lines is less than one. The number is determined by subtracting the valid number of OCUs from the valid number of POTs.
- *In point-to-point circuits* – the number of OCUs using "E" type lines is less than two. The number is determined by subtracting the valid number of OCUs from the valid number of POTs.
- *When comparing the "P" line's CLEI (OCU HECIG) information with the HECIG TO EQP table* – no match is found.

Full Name: ODB258E OCU OUT OF TRANSMISSION SEQUENCE

Data Source: NSDB first determines OCUs by examining "P" lines and comparing the entries with the HECIG to EQP Table. The system then determines the OCU's position with respect to the nearest facility (is it located at the A or the Z side of the carrier channel bank). Then, looking in the opposite direction, it must be next to a nondigital access point, loop facility, or a repeater. This message is passed when none of these conditions are met.

Full Name: ODB259E CIRCUIT CONTAINS TOO MANY OCUs

Data Source: NSDB checks for MJU assignment and segment information to determine the circuit configuration (multi–point or point-to-point). The NI lines are then counted and compared with the algorithm in the table below to determine the valid number of terminations. For all services, the valid number of OCUs are determined by subtracting the appropriate value (in the table) from the number of POTs detected on the record. This message is returned when the number of OCUs detected exceeds the valid number indicated in the table.

**Table 10–1.** Valid Number of OCUs, by Service

| SERVICE | VALID # OCUs |
|---|---|
| Point-to-Point | 2 |
| Multi-point Remote Leg | 1 |
| Seg. Point-to-Point | 1 |

Full Name: ODB260E CIRCUIT SHOULD NOT CONTAIN OCU RECORD(S)

Data Source: This message is passed when NSDB detects (via E type lines) OCUs on a cascading section of a Multi-Point circuit. The valid number of OCUs on a cascading section segment is zero.

Full Name: ODB261E CIRCUIT MISSING OCU-DP RECORD(S)

Data Source: This message is passed if NSDB has detected a SLC96 carrier on a circuit, and is not able to detect an OCU-DP. This validation is accomplished by comparing the entries on the "P" lines with the HECIG to EQP table.

Full Name: ODB262E NETWORK INTERFACE RECORD FOR POT NOT FIRST OR LAST

Data Source: This message will be passed under either of these circumstances:

— If, using "NI" lines, one POT is detected and it is not in either the first or the last position on the circuit details.

— If, using "NI" lines, two POTS are detected and they are not in the first and last positions on the circuit details.

Full Name: ODB263E NETWORK INTERFACE RECORD FOR POT NOT NEXT TO CXR OR ACC PT RECORD

Data Source: This message is passed if the POT record detected is not next to a digital access point or carrier.

Full Name: ODB264E CIRCUIT MISSING NETWORK INTERFACE RECORD FOR POT

Data Source: This message is returned if, when comparing the circuit record with any associated carrier record, NSDB detects more POTs on the carrier than there were NI lines on the circuit.

Full Name: ODB266E MORE THAN TWO NETWORK INTERFACE RECORDS FOR POT

Data Source: This message is passed if more than two POTS (NI lines) are detected on a circuit record.

Full Name: ODB267E CIRCUIT MISSING NETWORK INTERFACE RECORD/S

Data Source: NSDB determines terminations by NI or S lines. NSDB will pass this message if, on a point-to-point circuit, it detects less than two terminations.

Full Name: ODB268E INVALID RATE IN NETWORK INTERFACE RECORD(S)

Data Source: This message is passed when a blank is detected in the Rate field of the Network Interface (NI) line.

Full Name: ODB269E CIRCUIT CONTAINS TOO MANY TERMINATION RECORD(S)

Data Source: NSDB and ITS check for MJU assignment and segment information to determine if the circuit is a multi-point. The NI or S lines are then counted and compared with the algorithm in the table below to determine the valid number of terminations. This message is returned when the number of NI and/or S lines is greater than the valid number of terminations.

**Table 10–2.** Valid Number of Terms, By Service

| SERVICE | VALID # OF TERMS |
|---|---|
| Point-to-point | 2 |
| Multi-point | 1 |
| Cascading Section MP | 0 |
| Seg. point-to-point | 1 |

Full Name: ODB270E TERMINATION RECORD OUT OF TRANSMISSION SEQUENCE

Data Source: NSDB passes this message when, on a point-to-point circuit, the termination information (indicated by NI or S lines) is not the first and last positions of the circuit record.

Full Name: ODB271E LATS II OR DTAU NOT ACCOMPANIED BY ACCESS POINT DATA

Data Source: This message is passed when the LATS II hardware information is not found next to the access information.

Full Name: ODB272E POSSIBLE ACCESS POINT CONFIGURATION OR SEQUENCE ERROR

Data Source: NSDB detects access points by the presence of AP note lines. If the configuration is DAB or DBA, it does not want to be next to a repeater, an NI Sub-Rate line, a loop facility, or station. NSDB passes this message when the conditions for either configuration are not met.

Full Name: ODB273E INVALID CONFIGURATION ON TEST ACCESS RECORD

Data Source: Currently, NSDB passes this message if the configuration code ACCESS is *not* equal to "4AB," "4BA," "4WF," "DAB," or "DBA."

Full Name: ODB274E INVALID ORIENTATION ON TEST ACCESS RECORD

Data Source: NSDB passes this message if the orientation is not "EF" or "FE."

Full Name: ODB275E CARRIER IN REVERSE SEQUENCE TO POT RECORD

Data Source: NSDB compares the CLLI information on the circuit facility assignment lines (XA, XZ) with the ODB OFFICE DATA TTS table. If the system discovers a POT, and an NI line with a "15" or a blank is detected at the opposite end of this service, this message is returned.

Full Name: ODB276E CAN NOT RECOGNIZE TERMINATION POINT ON SEGMENTED PT-TO-PT CIRCUIT

Data Source: NSDB and ITS use NI, OCUs, LOOPs, and S lines to determine termination points on segmented point-to-point circuits. If a combination of this information is missing or contained within note lines, NSDB/ITS cannot recognize the termination points.

Full Name: ODB277E CIRCUIT ARRANGEMENT REFERENCES TOO MANY POT(S)

Data Source: NSDB and ITS check for MJU assignment and then segment information to determine if the circuit is a MULTI-POINT. The NI or S lines are then counted and compared with the algorithm in table below to determine the valid number of terminations.

**Table 10–3.** Valid Number of Terms, By Service

| SERVICE | VALID # OF TERMS |
|---|---|
| Point-to-point | 2 |
| Multi-point | 1 |
| Cascading Section MP | 0 |
| Seg. point-to-point | 1 |

Full Name: ODB278E ACCESS POINT NUMBERS ARE NOT UNIQUE OR NOT ASCENDING

Data Source: This message is passed if the access point numbers, after being checked in transmission order, are not unique or not in ascending order.

Full Name: ODB280E CONFIGURATION/ORIENTATION ERROR ON TEST ACCESS RECORD

Data Source: This information is returned if the combination of configuration and orientation information on a test access record is not in reverse alphabetical order (i.e., EF/4BA).

Full Name: ODB285E UNABLE TO PARSE UNIT NUMBER FIELD FOR PORT NUMBER

Data Source: The number after the dash in the unit number field is divided by six. Subtract one from the remainder to get the actual port number. This message is returned if this cannot be performed for some reason.

Full Name: ODB286E QUAD MJU LOCATION OR RELAY RACK ARE BLANK

Data Source: This message is returned if the location code or relay rack are blank, denoting missing assignments for Quad MJUs.

### 10.1.2 DS1/CLFI Data Validations

Full Name: ODB265E CIRCUIT CONTAINS TOO MANY NETWORK INTERFACE RECORD/S

Data Source: This message is returned when more than one NI line is detected on HICAP circuits with a CLFI format, or a Special Service circuit with a DH service code.

Full Name: ODB267E CIRCUIT MISSING NETWORK INTERFACE RECORD(S)

Data Source: HC service codes must have NI lines at both the A and the Z ends. CLFIs must have an NI line at either the A or the Z end.

Full Name: ODB273E INVALID CONFIGURATION ON TEST ACCESS RECORD

Data Source: Access point configuration is not 4AB or 4BA.

Full Name: ODB274E INVALID ORIENTATION ON TEST ACCESS RECORDS

Data Source: The access point Orientation is not not FE or EF.

Full Name: ODB278E ACCESS POINT NUMBERS ARE NOT UNIQUE OR NOT ASCENDING

Data Source: The access point numbers must be unique and in ascending order.

Full Name: ODB279E CIRCUIT MISSING STATION EQUIPMENT RECORD(S)

Data Source: DS1 circuits with DH service codes must have an S line at one or both ends of the circuit.

Full Name: ODB280E CONFIGURATION/ORIENTATION ERROR ON TEST ACCESS RECORD

Data Source: This message will be returned if the combination of Configuration and Orientation codes on the test access record is not in reverse alphabetical order (i.e., 4AB/FE).

Full Name: ODB281E NO ACCESS POINTS PRESENT ON THE CIRCUIT

Data Source: No access point information was found on the circuit.

Full Name: ODB282E ACCESS POINT NOT ACCOMPANIED BY ACCESS POINT DATA

Data Source: Access point equipment was found, but was not next to a TP record. Digital access equipment was found on the circuit (E lines); however, the expected access point information was not next to it.

Full Name: ODB283E CIRCUIT HAS TOO MANY STATION EQUIPMENT RECORD(S)

Data Source: This message is returned when more than two S lines are detected on a DS1 circuit with a DH service code.

Full Name: ODB284E INVALID LOCATION FORMAT ON NETWORK INTERFACE RECORD(S)

Data Source: The Location Format (CLLI) on the NI line is neither 8 or 11 characters in length, or the first three characters are not unique.

## 10.2 Appendix B – Summary of Features from Release 3.1

### 10.2.1 Enhanced Match Merge for Service Orders

The SOAC 18.6 release established Service Order and Work Order interfaces with NSDB. Included in the data sent to NSDB were various TIRKS fields to support dual provisioning flows for all designed orders. NSDB received but did not utilize all of this data in release 3.0. One restriction placed on the NSDB 3.0 was that it would only handle non-segmented, single-end TIRKS/designed specials over the SOAC-NSDB interface, and that it would not support change of order class (ADSR) or Wire Center using C/T lines on a Service Order. The reasons for these restrictions were that priority was given to the major features that were needed for the SOAK (e.g. ISDN, POTS/Non-designed flows, and the new NSDB Line Record). Now in NSDB 3.1 this data will be fully utilized, and all designed specials will be handled via the SOAC interface as well as the current TIRKS interface.

There are feature dependencies between the SOAC 18.6 Maintenance Release, WFA/C 1.1, and NSDB 3.1. To fully support Wire Center changes for designed circuits, NSDB needs additional fields that are provided in the SOAC 18.6 Maintenance Release. The enhanced match/merge in NSDB 3.1 could be used with a WFA/C 1.0 release level, but support for changing from a designed to a non-designed circuit (or vice versa) using the C/T lines require WFA/C 1.1 to be installed.

No new NSDB tables or table entries are required for this feature.

### 10.2.2 Enhanced Matching for Work Orders

The Work Order interface is a real-time application to application interface from SOAC to NSDB to include completed work order data received from the Loop Facilities Assignment and Control System (LFACS) and the Computer System for Mainframe Operations (COSMOS)/SWITCH. Non-circuit related changes are filtered out by SOAC and are not sent to NSDB.

When the Circuit Installation and Maintenance Assistance Package (CIMAP) Release 3.6.x is converted to WFA/C Release 1.0, every record in CIMAP will have data from the Circuit History Database and/or the Installation Administration Database that will be copied and/or moved into NSDB. The SOAC/NSDB Work Order Interface (WOI) is designed to maintain loop and office equipment assignments stored in the NSDB Line Record Database.

The Enhanced Matching for Work Orders feature in NSDB 3.1 resolves the problem of how to identify which special service circuit in NSDB is being updated by the Work Order message from SOAC. This problem exists because the circuit identifier used in SOAC (Circuit Termination Identifier, or CTID), never existed in the CIMAP databases.

The Enhanced Matching for Work Orders feature utilizes a new hierarchy of selective match criteria by which common attributes of a circuit ID can be checked for potential matches. Once a match is made by NSDB, the CTID data is stored and will be used to match all future SOAC Work Order messages.

This feature uses a new field in the ODB CO OPTIONS table. This field called MANUALWO is used to select whether or not unmatched SOAC Work Order messages will be stored along with their candidate circuit IDs for manual update activity. Another existing table, ODB WCLLI – WCTR may also be called during the matching process. This table is used to map the outside plant facility wire center CLLI to its SOAC wire center value.

In addition to the tables mentioned above, a new user format in WFA/C is available for manual resolution of failed matches. This WFA/C format is called OSSMFA, or the Match Failure Administration screen.

A new NSDB database has been created to store the SOAC work order messages that fail matching, along with their potential circuit ID match candidates. This partitioned data base is called the Match Failure Database, VJDBMFDD. This NSDB database can be accessed by the OSSMFA format.

### 10.2.3 Service Order Image Database

As of SOAC Release 18.6, the Service Order and Assignment Images were sent to NSDB 3.0. These images were not stored in NSDB. The Service Order Image feature for NSDB 3.1 provides a new NSDB database to store the Service Order Images. Access to this data is via WFA/C format OSSIMG, the Service Order Image screen.

An archive run, VJARM01, is now available to periodically remove these images from the NSDB database. The archive run is user defined, and is based on a time interval measured from the completion date of the order. This run also allows for deletions of the images in addition to archival.

The NSDB system administrator must make entries on the existing ODB IMAGE table to indicate the Section identifiers that NSDB will encounter on the Service Order. The IMG_DB field is used to control loading of the Image database.

### 10.2.4 Hunt Group Database

As of SOAC 18.6, Hunt Group information was sent to NSDB 3.0. This data was not stored is a separate database, but the information was stored with each individual circuit. This meant that the WFA/C users could view a Line Record and see that the circuit was part of a hunt group, but could not view the data associated with an entire group (e.g. all terminals in the group, call sequence, etc.).

The Hunt Group feature for NSDB 3.1 provides a separate Hunt Group database in NSDB, and allows user access to view the data associated with an entire group. The WFA/C formats are OSSLR and OSSLRH, or the Line Record screen and the Line Record Hunt screen. No new NSDB tables or table entries are required for this feature.

### 10.2.5 OQS for Designed Circuits

In NSDB 3.1, the designed circuit databases and the new Hunt Group database are accessible via the Open Query System (OQS). The designed circuit databases are; Ineffect Circuit Carrier Database, Pending Circuit Carrier Database, Message Circuit Carrier Database, Message Circuit Index Database, Multi-point Database, and the Circuit Index Manager Database.

For more information on this feature, refer to Bellcore BR 190-534-320, *How to Use OQS for NSDB.*

### 10.2.6 Multi-Machine OQS

The new Multi-Machine feature of OQS allows users to run queries against databases that reside on remote IMS systems. Currently, NSDB, WFA/C, WFA/DI, and WFA/DO are the systems that can use this feature. Communication between the IMS systems is accomplished by using Multiple System Coupling (MSC) links. A new S1 security grid, OQSGEN, is used to limit access to Multi-Machine queries. Successful Multi-Machine queries require security permission by both systems, and the establishment of the MSC links. For more information about the Multi-Machine OQS feature, see the Bellcore BR 190-534-300, *NSDB Open Query System (OQS) User Manual.*

### 10.2.7 Enhanced Message and Carrier Maintenance Support for WFA/C

This enhancement is provided to help WFA/C satisfy some of the FCC Docket 88–2 requirements for reporting maintenance activities on selected Message and Carrier services. To accomplish this, WFA/C needed enhancements to handle and control trouble reports on Message and Carrier circuits.

NSDB 3.1 provides WFA/C with the data necessary to perform Trouble Ticket Administration for Special, Message, and Carrier circuits. This information includes enhanced partial ID queries by the non-designed circuit ID, and translation data for additional switch types.

The trouble ADD process in WFA/C may be triggered by the NMA system, or directly from a user at a WFA/C terminal. The resulting query to NSDB comes from WFA/C via a dynamic link in either case.

The NSDB system administrator must make entries on the new ODB MON SWITCH table so that the translation data for all switch types is available to WFA/C. Another NSDB table, ODM EXK TO SW ID, also requires entries to support ISDN queries triggered by NMA. This query is invoked by WFA/C when NMA detects a trouble on a ISDN line and only the OE and switch CLLI are passed to WFA/C from NMA as query keys.

### 10.2.8 The SOAC 18.6 Maintenance Release Enhancements to NSDB

The following enhancements to the SOAC/NSDB service order interface will be available with the installation of NSDB 3.1, the SOAC 18.6 Maintenance Release, and WFA/C 1.1.

*Routing by MCN*

The Major Customer Number (MCN) field identifier can be used to trigger NSDB involvement with a service order as of NSDB 3.0 and SOAC 18.6. The enhancement will enable selective involvement based on the MCN field value. This will allow only desired MCN candidates to be passed to NSDB. This enhancement is invoked through SOAC table settings, and no additional NSDB processing is required.

*Processing Changes Using C & T Action Lines*

A new tag, CTPRNUM, is transmitted from SOAC to NSDB to aid NSDB in the processing of multi-wire center service orders where the circuit termination identifier changes across service order C & T action lines.

For more information on this tag, and the SOAC processing changes, see TM-STS-019846, *SOAC Specifications for the SOAC/NSDB (Network and Services Data Base) Interface – Issue 6.*

### 10.2.9 NSDB Load of LEIS extract data from LFACS

A new run, VJCDU16, is provided to load, refresh, and update all loop information from the LFACS source without the need for a complicated external process.

### 10.2.10 ISDN Support for NMA/Switch

NSDB 3.0.3 provided ISDN support for NMA via FCIF contracts. Data is sent from NSDB to NMA upon the following events:

- Completions of Line and Station Transfers (LST) from SOAC.
- Completions of ISDN circuits from WFA/C.
- Completion of ISDN circuits from SOAC if WFA/C is not used.

- Updates of the in-effect circuits from SOAC and WFA/C.
- Telephone number swaps from a CRS system.

The NSDB system administrator must make entries, or verify selected field values in the following NSDB tables to support this feature; ODB CO OPTIONS, ODB DATA ROUTING, ODB ROUTEBY, ODB SERVICE CODE, and ODM EXK TO SW ID.

### 10.2.11 Archive Feature

This feature applies to NSDB's Line Record, Service Order Image, and Audit databases. New online formats are provided for entry of control card information used in the archive run, VJARM01. (See Runbook BR 190-534-058 for more information.) The control card information includes the aging criteria used to identify archive candidates.

### 10.2.12 Bulk Update

This feature enables NSDB system administrators to change the value of specified fields for a large number of records in NSDB databases. The candidate records for the bulk update can be provided through an OQS query, or received from an external system. The NSDB databases that can be bulk updated are the Circuit Carrier In-Effect, and Circuit Carrier Pending databases, and the Message, Line Record, and Multi-point databases.

This feature is provided to allow for the update of large numbers of records that require changes when the normal updating flows do not apply. This functionality currently exists in both WFA/C and the TIRKS system.

The bulk runs VJOQR04, VJCDU17, and VJCDU20 are used in the Bulk Update feature.

### 10.2.13 SS7 Support in NSDB

In order to support Signaling System 7 (SS7) usage in an inter-LATA environment, the TIRKS system added 4 new fields on the TIRKS WORD-CD. NMA/Switch may require these fields in the future if the BCCs decide to tariff the monitoring of SS7 networks to the Inter-exchange Carrier switches. In NSDB 3.1 the Message Circuit database has added the 4 new fields from the WORD-CD. These fields are:

- TCIC Trunk Circuit Identification Code
- POINTCDA Point Code for A end of the circuit
- POINTCDZ Point Code for Z end of the circuit
- CUSLIT Literal value of Customer Location if in CLLI format

This feature is provided in anticipation of NMA/Switch monitoring of these circuits.

### 10.2.14 Billing Attribute Support in NSDB

This feature in NSDB is the result of a TIRKS system enhancement. The TIRKS system has been enhanced (as of release 16.0) to allow for the removal of data currently stored in the billing attributes TSP, AVN, ESV, DIV, and DV from in-effect circuits in TIRKS. WFA/C and NSDB must also have this ability for these fields in their databases.

This feature will be invoked in NSDB upon receipt of the appropriate SOAC service order message. This message will concurrently trigger TIRKS and WFA/C to remove the data in their databases.

### 10.2.15 ISDN Testing Using the New WFA/DO-NSDB-ITS/ISDN Interface

Maintenance testing of Basic Rate ISDN service from WFA/DO is now possible with this new interface. NSDB stores the line record data required by ITS for maintenance testing Basic Rate ISDN services. Three new TCM paths are required between WFA/DO and NSDB to support this interface. The paths from NSDB to ITS/ISDN used in release 3.0 for the WFA/C-NSDB-ITS/ISDN interface are also used in the new WFA/DO-NSDB-ITS/ISDN interface.

### 10.2.16 Three New NSDB Tables

The first new NSDB table supports data routing for switch types that are required for WFA/C and/or NMA/Switch. This table, ODB MON SWITCH, is described in chapter 7.

```
COMMAND          ***  NSDB-TTS DATA SCREEN  ***                    /FOR


TABLE NAME:  ODB MON SWITCH    TABLE KEY: NMASWI              ADMIN AREA:
TABLE RECORD KEY:  DMC                          # OF RECORDS:
NOTE: MONITORED SWITCH TYPES                   REL/LEV:            MOD: Y
   FIELD     FIELD
   NAME      VALUE
   UNIV_CD   DMC*









TTS210I FIND COMPLETED
```

**Figure 10–1.** ODB MON SWITCH TTS Table