

Functional Overview Of Con-nect SNADS

This chapter provides a functional overview of Con-nect SNADS, which can be either run under Com-plete using the EntireX Broker Services LU6.2 API (referred to in this chapter as the LU6.2 API), batch mode and the LU6.2 API, or CICS.

It covers the following topics:

- Con-nect SNADS in Com-plete
- Con-nect SNADS in Batch Mode
- Con-nect SNADS in CICS

Note:

Con-nect SNADS cannot handle documents larger than 1 MB.

Con-nect SNADS in Com-plete

The following topics are covered below:

- Con-nect SNADS Concepts
- Con-nect SNADS and the SNADS Network
- Con-nect SNADS and Con-nect
- Con-nect SNADS and Natural
- Con-nect SNADS and Natural Security

Con-nect SNADS Concepts

Con-nect SNADS is an implementation of the IBM SNA Distribution Services (SNADS) architecture. The implementation of Con-nect SNADS in Com-plete with the LU6.2 API provides the support for application-to-application communication between Con-nect and other SNADS nodes. Con-nect SNADS is written primarily in Natural and invokes the communication services of the LU6.2 API with the Natural `PROCESS` statement.

Con-nect SNADS implements SNA Distribution Services functions by means of a program or set of programs called "queue servers". A queue is an area where Distribution Interchange Units (DIUs) reside while awaiting further processing by a queue server. The Con-nect SNADS programs `DS_RECEIVE`, `DS_ROUTER_DIRECTOR`, and `DS_SEND` process DIUs. The following is a brief description of each program:

DS_RECEIVE

The `DS_RECEIVE` program processes DIUs which are received from an adjacent node and inserts them into the inbound queue of the current node. A Con-nect SNADS "demon" process continuously listens for conversation requests from remote SNADS partner nodes. When a request is caught, the demon process executes the `DS_RECEIVE` program as an inline subroutine.

The demon process itself runs as an attached task in the Com-plete environment and uses a dedicated Con-nect SNADS queue called the dummy queue for task control purposes.

The Con-nect SNADS supplied startup routine, `CPSEND`, leads the flow of control into the appropriate Natural environment.

DS_ROUTER_DIRECTOR

If the recipients of the DIUs are elements (addressees) of a remote SNADS node, the router component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into one or several of the local system's outbound queues.

Whereas, if the recipients of the DIUs are elements (addressees) of the local Con-nect system, the director component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into the appropriate multi-node routine. This routine delivers the DIUs into the recipients' Inbasket.

DS_ROUTER_DIRECTOR is invoked by Con-nect SNADS with the Complete ATTACH function. The Con-nect SNADS supplied startup routine, CPSEND, leads the flow of control into the appropriate Natural environment.

DS_SEND

The DS_SEND program processes DIUs from an outbound queue of the current node into the DS_RECEIVE program of an adjacent node.

Con-nect SNADS invokes the DS_SEND process with the Complete ATTACH function. The Con-nect SNADS supplied startup routine, CPSEND, leads the flow of control into the appropriate Natural environment.

All Con-nect SNADS nodes have one and only one inbound queue. This queue is used for both inbound DIUs and DIUs which originate from the local Con-nect system.

All DIUs make their initial entry to, or begin their departure from, the Con-nect node through the inbound queue. Thus, there is a common starting point for Con-nect to invoke SNADS functions.

Installations also have one or several outbound queues which are used to direct DIUs to other nodes. In addition to the outbound queues, installations have a dummy queue which is used to control the execution of the demon process.

Con-nect users can exchange information with users on other Con-nect nodes and with users of other office systems that provide SNADS implementations (e.g. DISOSS). Additionally, Con-nect users can also exchange information with users of office systems that do not provide SNADS implementations (e.g. PROFS) whenever gateway functionalities such as those provided by DISOSS or Soft-Switch Central are used.

Con-nect SNADS and the SNADS Network

You need to choose a suitable location for Con-nect SNADS in your SNADS network by determining which existing SNADS node(s) the new system will communicate with directly. Con-nect SNADS should then be located adjacent to that node.

Since the existence and location of the Con-nect SNADS must be properly reflected in the other SNADS network nodes (i.e. in their routing tables), ensure that the Con-nect SNADS location you have chosen can be reasonably implemented in the SNADS routing tables of both existing SNADS nodes and the new Con-nect SNADS.

Con-nect SNADS uses SNA logical unit type 6.2 (LU6.2) links to exchange data with neighboring nodes in the SNADS network. These links must be established before Con-nect SNADS can be used.

If routing tables are not consistently specified, DIUs may be forwarded endlessly through the SNADS network. To prevent this from occurring, the SNADS architecture uses a "hop count" to limit node-to-node forwarding operations.

The hop count is specified in SNADS Distribution Interchange Units (DIUs) and defines the maximum number of "legs" along which the DIU can travel. Con-nect SNADS uses a value specified at installation (though it can be changed later) as the hop count for all DIUs that originate from this location.

Con-nect SNADS and Con-nect

Each Con-nect SNADS system uses one Con-nect spool file as a data repository. However, that spool file can service more than one Con-nect system file.

Note:

Each Con-nect system file serviced by the Con-nect SNADS system must be assigned at least one SNADS Distribution Service Unit Name (DSUN) or SNADS "node name".

Since Con-nect SNADS uses features of multi-node Con-nect internally to convey information between the Con-nect spool file and the associated Con-nect system file(s), each Con-nect system file must have a unique multi-node node name stored in the spool file.

Note:

Node names are stored in the spool file using the Initialize Node File function of the Con-nect ADMIN program.

The Con-nect system file node name (the name specified as your Con-nect system during the Initialize Node File function) is used as the Distribution Group Name (DGN) along with the Con-nect user ID (as Distribution Element Name (DEN)) to form the default Distribution User Name (DUN):

```
DUN=DGN.DEN  
DUN=node.user
```

Con-nect SNADS and Natural

The size of the Natural buffer pool and the number of Complete threads should reflect the increased system load that will be created by the Con-nect SNADS routines.

Requirements for the Natural Nucleus Parameter (NATPARM)

As part of the installation process, you will need to ensure that certain Natural parameters are set correctly. For a listing of the Natural parameters, see *Modify the "queue server" front-end parameter module CONSNADZ*. Con-nect SNADS can use the Natural nucleus with either of the following setups:

1. In the "end user" application of Con-nect (i.e. the Send function).
2. In the "server programs" application of Con-nect SNADS (e.g. the DS_SEND program).

The setups above impose a number of requirements for the NATPARM settings. However, the requirements for the second setup are more specific and restrictive.

Many sites will find the required NATPARMs for the "server programs" application unacceptable in normal Natural operations. Depending upon how the Natural nuclei are used, there are three options for where and when you can set the required NATPARMs:

1. You can use one nucleus for both the "server programs" application and the "end user" application with a NATPARM that enforces the restrictive "server usage" requirements.
2. You can choose to generate a separate nucleus solely for the use of the "server programs" application.
3. You can use one nucleus for both the "end user" and "server programs" applications with a NATPARM that enforces only the less restrictive "end user" requirements and the Natural session parameters to enforce the more restrictive "server programs" requirements.

This example can be set up by means of Natural dynamic parameters at the beginning of the respective Natural sessions. The dynamically set NATPARMs can be specified in the respective keywords in the CONSNADZ module (e.g. DYNRECV).

The CONSNADZ Parameter Module

The CONSNADZ module is linked with the front-end programs, CPSEND, CPINIT, CPTERM, and the back-end program CPEND. The purpose of these 370 assembler programs is twofold:

- They permit the Natural "server programs" application to be invoked by a single Complete transaction code;
- They permit Natural dynamic parameters to be passed to the nucleus to be used by the "server programs" application.

If required, CPDBAS can be linked to the front- and back-end programs so that Adabas OPEN (OP command) and CLOSE (CL command) commands occur at the beginning and end of an asynchronous Natural task.

You can satisfy the NATPARM requirements by using dynamic overrides specified within the CONSNADZ module. This eliminates the need for a separate nucleus while satisfying the NATPARM requirements of the "server programs" application.

For information about the CONSNADZ parameter module see *Modify the "queue server" front-end parameter module CONSNADZ*.

Note:

Since errors in this portion of the system are difficult to debug, it is recommended that you determine the method you will use to satisfy the NATPARM requirements before you begin the installation process. Any modifications to the Natural dynamic parameters should be verified in the interactive mode.

TP Monitor and Natural Nucleus Considerations

Con-nect SNADS requires LU6.2 support for the physical transmission of messages and documents between nodes. The Com-plete ATTACH function invokes the Con-nect SNADS "server programs" application. Therefore the "server programs" application of Con-nect SNADS must be executed by Natural running under Com-plete.

Since the "end user" application does not perform the physical transmission of data, you can configure your Con-nect SNADS system so that users are supported from a different TP monitor even though the "server programs" application runs from Natural under Com-plete.

This can be achieved by using a single Con-nect spool file that is shared by the "server programs" application under Com-plete and the "end user" application. The "end user" application can run in another Com-plete or a different system such as CICS. Any TP monitor which is supported by the installed Natural version is acceptable for the "end user" application.

For this configuration alternative, the following are required:

- Two Natural nuclei, one for the Com-plete environment and the other for the "end-user" environment.
 - The Com-plete Natural nucleus must satisfy all installation requirements for the "server programs" application.
 - The Natural nucleus on the "end user TP monitor" must satisfy all installation requirements for the "end user" application.
- Optionally, a single Natural system file (FNAT) can be shared between the nuclei.

In addition, the following requirements must be satisfied for the Con-nect system files and spool (telex) file:

- A single Con-nect "spool" (telex) file must be shared between the Natural nuclei (logical file number 223 in the NATPARM) for the "end user" and "server programs" applications.
- A link between the Con-nect system file(s) and the shared Con-nect spool file must be created. The following is a description of how this link can be created:
 - For each Con-nect system file you have, a "home node record" must be defined to the spool file using the Initialize Node File function on the Con-nect "Administration - External Mail Nodes" screen. For further information on the Initialize Node File function, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance*, sub-section *Define Local Node*. This process creates a record with a name (used as the DGN for SNADS), the current physical DBID, and the current physical FNR of the logical file number 251.
 - If separate logical 251 files are used, you must invoke Natural with a different logical file 251 setting each time you write these records in order to properly value the home node record. This can be achieved by using the Natural dynamic parameter LFILE.
 - If you move a Con-nect system file to a different database or file number, you must re-initialize the "home node record" with the new setting before mail can be sent or received.
 - You can verify the correlation of DSUNs to DGNs and their physical locations (DBID,FNR) by selecting the Control Maintenance function on the Con-nect "SNADS Administration Menu" screen. For further information, see the *Con-nect Administration* documentation, section *Con-nect SNADS*, sub-section *Control Maintenance*.

Con-nect SNADS and Natural Security

Natural assigns a user ID to both non-interactive and interactive sessions. By default, Natural passes the current user ID to the Adabas nucleus. The user ID allows Adabas to access any end of transaction (ET) data that can be stored for that particular user ID. However, Adabas cannot accept simultaneous sessions with the same user ID. This restriction can cause problems for the DS_RECEIVE, DS_ROUTER_DIRECTOR, and DS_SEND programs. If Natural Security is implemented, more than one instance of the programs can be active simultaneously and all of the programs may attempt to logon with the same Natural Security user ID.

To avoid conflicts resulting from more than one active session assigned to the same user ID, assign a value of blanks to the Natural keyword ETID, i.e. ETID= ' ' (blank). Since the Con-nect SNADS server programs do not use the ET data facility, data integrity problems will not occur.

Note:

The ETID setting can be modified in the Natural Security definition or in the dynamic parameter settings in CONSNADS.

Additionally, Natural does not issue OP and CL commands at the beginning or end of a session whenever ETID is set to blanks. Nevertheless, the module CPDBAS can be linked to the front- and back-end programs to issue OP and CL commands against databases which are specified in the CONSNADZ parameter module.

Con-nect SNADS in Batch Mode

The following topics are covered below:

- Con-nect SNADS Concepts
- Con-nect SNADS and the SNADS Network
- Con-nect SNADS and Con-nect
- Con-nect SNADS and Natural

Con-nect SNADS Concepts

Con-nect SNADS is an implementation of the IBM SNA Distribution Services (SNADS) architecture. Software AG's LU6.2 API provides the support for application-to-application communication between Con-nect and other SNADS nodes. Con-nect SNADS, which is written in Natural, invokes the communication services of the LU6.2 API with the Natural PROCESS statement.

Con-nect SNADS implements SNA Distribution Services functions by means of a program or a set of programs called "queue servers". A queue is an area where Distribution Interchange Units (DIUs) reside while awaiting further processing by a queue server. The Con-nect SNADS programs DS_RECEIVE, DS_ROUTER_DIRECTOR, and DS_SEND process DIUs. The following is a brief description of each program:

DS_RECEIVE

The DS_RECEIVE program processes DIUs which are received from an adjacent node and inserts them into the inbound queue of the current node. A Con-nect SNADS "demon" process continuously listens for conversation requests from remote SNADS partner nodes. When a request is caught, the demon process executes the DS_RECEIVE program as an inline subroutine.

The demon process itself runs in batch mode where it exclusively occupies a continuously executing Natural task called the "input handler" and uses a dedicated Con-nect SNADS queue called the dummy queue for task control purposes.

DS_ROUTER_DIRECTOR

If the recipients of the DIUs are elements (addressees) of a remote SNADS node, the "router" component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into one or several of the local system's outbound queues.

Whereas, if the recipients of the DIUs are elements (addressees) of the local Con-nect system, the "director" component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into the appropriate multi-node routine. This routine delivers the DIUs into the recipients' Inbasket.

DS_ROUTER_DIRECTOR is executed by Con-nect SNADS with a continuously running Natural task called the "queue server". The DS_ROUTER_DIRECTOR can share this batch task with the DS_SEND program of one or multiple outbound queues but not with the demon process.

DS_SEND

The DS_SEND program processes DIUs from an outbound queue of the current node into the DS_RECEIVE program an adjacent node.

Con-nect SNADS executes the DS_SEND process in a continuously running Natural task called the "queue server". The DS_SEND can share this batch task with the DS_ROUTER_DIRECTOR program of one or multiple outbound queues but not with the demon process.

All Con-nect SNADS nodes have one and only one inbound queue. This queue is used for both inbound DIUs and DIUs which originate from the local Con-nect system.

All DIUs make their initial entry to, or begin their departure from, the Con-nect node through the inbound queue. Thus, there is a common starting point for Con-nect to invoke SNADS functions.

Installations also have one or several outbound queues which are used to direct DIUs to other nodes. In addition to the outbound queues, installations have a dummy queue which is used to control the execution of the demon process.

Con-nect users can exchange information with users on other Con-nect nodes and with users of other office systems that provide SNADS implementations (e.g. DISOSS). Additionally, Con-nect users can also exchange information with users of office systems that do not provide SNADS implementations (e.g. PROFS) when gateway functionalities such as those provided by DISOSS or Soft-Switch Central are used.

Con-nect SNADS and the SNADS Network

You need to choose a suitable location for Con-nect SNADS in your SNADS network by determining which existing SNADS node(s) the new system will communicate with directly. Con-nect SNADS should then be located adjacent to that node.

Since the existence and location of the Con-nect SNADS must be properly reflected in the other SNADS network nodes (i.e. in their routing tables), ensure that the Con-nect SNADS location you have chosen can be reasonably implemented in the SNADS routing tables of both existing SNADS nodes and the new Con-nect SNADS.

Con-nect SNADS uses SNA logical unit type 6.2 (LU6.2) links to exchange data with neighboring nodes in the SNADS network. These links must be established before Con-nect SNADS can be used.

If routing tables are not consistently specified, DIUs may be forwarded endlessly through the SNADS network. To prevent this from occurring, the SNADS architecture uses a "hop count" to limit node-to-node forwarding operations.

The "hop count" is specified in SNADS Distribution Interchange Units (DIUs) and defines the maximum number of "legs" along which the DIU can travel. Con-nect SNADS uses a value specified at installation (though it can be changed later) as the "hop count" for all DIUs that originate from this location.

Con-nect SNADS and Con-nect

Each Con-nect SNADS system uses one Con-nect spool file as a data repository. However, that spool file can service more than one Con-nect system file.

Note:

Each Con-nect system file serviced by the Con-nect SNADS system must be assigned at least one SNADS Distribution Service Unit Name (DSUN) or SNADS "node name".

Since Con-nect SNADS uses features of multi-node Con-nect internally to convey information between the Con-nect spool file and the associated Con-nect system file(s), each Con-nect system file must have a unique multi-node node name stored in the spool file.

Note:

Node names are stored in the spool file using the Initialize Node File function of the Con-nect ADMIN program.

The Con-nect system file node name (the name specified as your Con-nect system during the Initialize Node File function) is used as the Distribution Group Name (DGN) along with the Con-nect user ID (as Distribution Element Name (DEN)) to form the default Distribution User Name (DUN):

```
DUN=DGN.DEN  
DUN=node.user
```

Con-nect SNADS and Natural

The size of the Natural buffer pool should reflect the increased system load that will be created by the Con-nect SNADS routines.

Requirements for the Natural Nucleus Parameter (NATPARM)

As part of the installation process, you will need to ensure that certain Natural parameters are set correctly. For a listing of the Natural parameters, see *Installing Con-nect SNADS in Batch Mode*. Con-nect SNADS can use the Natural nucleus with either of the following setups:

1. In the "end user" application of Con-nect (i.e. the Send function).
2. In the "server programs" application of Con-nect SNADS (e.g. the DS_SEND program).

The setups above impose a number of requirements for the NATPARM settings. However, the requirements for the second example are more specific and restrictive.

Many sites will find the required NATPARMs for the "server programs" application unacceptable in normal Natural operations. Depending upon how the Natural nuclei are used, you have two options for where and when you can set the required NATPARMs:

1. You can choose to generate a separate Natural batch nucleus solely for the use of the "server programs" application.
2. You can specify the more restrictive parameters for the "server programs" application as dynamic overrides in the z/OS JCL.

TP Monitor and Natural Nucleus Considerations

Con-nect SNADS requires SNA LU6.2 support for the physical transmission of messages and documents between nodes. Additionally, two batch tasks, the "input handler" and the "queue server" are required to schedule the "server programs". Therefore the "server programs" application of Con-nect SNADS must be executed by Natural running in batch mode.

Because the "end user" application does not perform the physical transmission of data, you can configure your Con-nect SNADS system so that users are supported from any TP monitor even though the "server programs" application runs from Natural in batch mode.

This can be achieved by using a single Con-nect spool file that is shared by the "server programs" application in batch mode and the "end user" application. The "end user" application can run in a TP system such as Com-plete.

For this configuration alternative, the following are required:

- Two Natural nuclei, one for the batch tasks and the other for the "end-user" environment.
 - The Natural batch task nucleus must satisfy all installation requirements for the "server programs" application.
 - The Natural nucleus on the "end user TP monitor" must satisfy all installation requirements for the "end user" application.
- Optionally, a single Natural system file (FNAT) can be shared between the nuclei.

In addition, the following requirements must be satisfied for the Con-nect system files and spool (telex) file:

- A single Con-nect "spool" (telex) file must be shared between the Natural nuclei (logical file number 223 in the NATPARM) for the "end user" and "server programs" applications.
- A link between the Con-nect system file(s) and the shared Con-nect spool file must be created. The following is a description of how this link can be created:
 - For each Con-nect system file you have a "home node record" must be defined to the spool file using the Initialize Node File function on the Con-nect "Administration - External Mail Nodes" screen. For further information on the Initialize Node File function, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance*, sub-section *Define Local Node*. This process creates a record with a name (used as the DGN for SNADS), the current physical DBID, and the current physical FNR of the logical file number 251.
 - If separate logical 251 files are used, you must invoke Natural with a different logical file 251 setting each time you write these records in order to properly value the home node record. This can be achieved by using the Natural dynamic parameter LFILE.
 - If you move a Con-nect system file to a different database or file number, you must re-initialize the home node record with the new setting before mail can be sent or received.
 - You can verify the correlation of DSUNs to DGNs and their physical locations (DBID,FNR) by selecting the Control Maintenance function on the Con-nect "SNADS Administration Menu" screen. For further information about the Control Maintenance function, see the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS*, sub-section *Control Maintenance*.

Con-nect SNADS in CICS

The following topics are covered below:

- Con-nect SNADS Concepts
- Con-nect SNADS and the SNADS Network
- Con-nect SNADS and Con-nect
- Con-nect SNADS and Natural
- Con-nect SNADS and Natural Security

Con-nect SNADS Concepts

Con-nect SNADS is an implementation of the IBM SNA Distribution Services (SNADS) architecture. The implementation of Con-nect SNADS under the CICS runtime environment is written in Natural and IBM 370 Assembler. CICS provides the support for application-to-application communication between Con-nect and other SNADS nodes. Con-nect SNADS invokes the communication services of CICS with a 370 Assembler language routine using CICS commands to access the corresponding functions provided by CICS.

Con-nect SNADS implements SNA Distribution Services functions by means of a program, or a set of programs, called "queue servers". A queue is an area where the Distribution Interchange Units (DIUs) reside while awaiting further processing by a "queue server". The Con-nect SNADS programs DS RECEIVE, DS_ROUTER_DIRECTOR, and DS SEND process DIUs. The following is a brief description of each function:

DS_RECEIVE

The DS_RECEIVE program processes DIUs which are received from an adjacent node and inserts them into the inbound queue of the current node. DS_RECEIVE is invoked by CICS whenever an incoming conversation request is received from a remote partner. The Con-nect SNADS supplied startup routine, CSRECV, leads the flow of control into the appropriate Natural environment.

DS_ROUTER_DIRECTOR

If the recipients of the DIUs are elements (addressees) of a remote SNADS node, the "router" component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into one or several of the local system's outbound queues.

Whereas, if the recipients of the DIUs are elements (addressees) of the local Con-nect system, the "director" component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into the appropriate multi-node routine. This routine delivers the DIUs into the recipients' Inbasket.

Con-nect SNADS invokes the DS_ROUTER_DIRECTOR program by means of the CICS interval control START command. The Con-nect SNADS supplied startup routine, CSSEND, leads the flow of control into the appropriate Natural environment.

DS_SEND

The DS_SEND program processes DIUs from an outbound queue of the current node into the DS_RECEIVE program of an adjacent node.

Con-nect SNADS invokes the DS_SEND process by means of the CICS interval control START command. The Con-nect SNADS supplied startup routine, CSSEND, leads the flow of control into the appropriate Natural environment.

All Con-nect SNADS nodes have only one inbound queue. This queue is used for both inbound DIUs and DIUs which originate from the local Con-nect system.

All DIUs make their initial entry to, or begin their departure from, the Con-nect node through the inbound queue. Thus, there is a common starting point for Con-nect to invoke the SNADS functions.

Installations also have one or several outbound queues. These queues are used to direct DIUs to other nodes.

Con-nect users can exchange information with users on other Con-nect nodes and with users of other office systems that provide SNADS implementations (e.g. DISOSS). Additionally, Con-nect users can also exchange information with users of office systems that do not provide SNADS implementations (e.g. PROFS) whenever gateway functionalities such as those provided by DISOSS or Soft-Switch Central are used.

Con-nect SNADS and the SNADS Network

You need to choose a suitable location for Con-nect SNADS in your SNADS network by determining which existing SNADS node(s) the new system will communicate with directly. Con-nect SNADS should then be located adjacent to that node.

Since the existence and location of Con-nect SNADS must be properly reflected in the other SNADS network nodes (i.e. in their routing tables), ensure that the Con-nect SNADS location you have chosen can be reasonably implemented in the SNADS routing tables of both existing SNADS nodes and the new Con-nect SNADS.

Con-nect SNADS uses SNA logical unit type 6.2 (LU6.2) links to exchange data with neighboring nodes in the SNADS network. These links must be established before Con-nect SNADS can be used.

If routing tables are not consistently specified, DIUs may be forwarded endlessly through the SNADS network. To prevent this from occurring, the SNADS architecture uses a "hop count" to limit node-to-node forwarding operations.

The "hop count" is specified in the SNADS Distribution Interchange Units (DIUs) and defines the maximum number of "legs" the DIU can travel. Con-nect SNADS uses a value specified at installation (though it can be changed later) as the "hop count" for all DIUs that originate from this location.

Con-nect SNADS and Con-nect

Each Con-nect SNADS system uses only one Con-nect spool file as a data repository. However, that spool file can service more than one Con-nect system file.

Note:

Each Con-nect system file serviced by the Con-nect SNADS system must be assigned at least one SNADS Distribution Service Unit Name (DSUN) or SNADS "node name".

Because Con-nect SNADS uses features of multi-node Con-nect internally to convey information between the Con-nect spool file and the associated Con-nect system file(s), each Con-nect system file must have a unique multi-node node name stored in the spool file.

Note:

Node names are stored in the spool file using the Initialize Node File function of the Con-nect ADMIN program.

The Con-nect system file node name (the name specified as your Con-nect system during the Initialize Node File function) is used as the Distribution Group Name (DGN) along with the Con-nect user ID (as Distribution Element Name (DEN)) to form the default Distribution User Name (DUN):

```
DUN=DGN.DEN  
DUN=node.user
```

Con-nect SNADS and Natural

The size of the Natural buffer pool and the number of Natural threads and Natural roll files should reflect the increased system load that will be created by the Con-nect SNADS routines.

Requirements for the Natural Nucleus Parameter (NATPARM)

As part of the installation process, you will need to ensure that certain Natural parameters are set correctly. For a listing of the Natural parameters, see *Modify the "queue server" front-end parameter module CONSNADS*. Con-nect SNADS can use the Natural nucleus with either of the following setups:

1. In the "end user" application of Con-nect (i.e. the Send function).
2. In the "server programs" application of Con-nect SNADS (e.g. the DS_SEND program).

The setups above impose a number of requirements for the NATPARM settings. However, the requirements for the second example are more specific and restrictive.

Many sites will find the required NATPARMs for the "server programs" application unacceptable in normal Natural operations. Depending upon how the Natural nuclei are used, you have three options for where and when you can set the required NATPARMs:

1. You can use one nucleus for both the "server programs" application and the "end user" application with a NATPARM that enforces the most restrictive "server usage" requirements.
2. You can choose to generate a separate nucleus solely for the use of the "server programs" application.
3. You can use one nucleus for both the "end user" and "server programs" applications with a NATPARM that enforces only the less restrictive "end user" requirements and the Natural session parameters that enforces the restrictive "server programs" requirements.

This can be set up by means of Natural dynamic parameters at the beginning of the respective Natural sessions. The Natural session parameters can be specified in the respective keywords in the CONSNADS module (e.g. DYNRECV).

The CONSNADS Parameter Module

The CONSNADS module is linked with the front-end programs, CSSEND, CSRECV, CSINIT, and CSTERM. The purpose of these 370 assembler programs is twofold:

- They permit the Natural "server programs" application to be invoked by a single CICS transaction code;
- They permit Natural dynamic parameters to be passed to the nucleus to be used by the "server programs" application.

If required, CSDBAS can be linked to the front- and back-end programs so that Adabas OPEN (OP command) and CLOSE (CL command) commands occur at the beginning and end of an asynchronous Natural task.

You can satisfy the NATPARM requirements by using dynamic overrides which are specified within the CONSNADS module. This eliminates the need for a separate nucleus while satisfying the NATPARM requirements of the "server programs" application.

For detailed information about the CONSNADS parameter module, see *Modify the "queue server" front-end parameter module CONSNADS*.

Note:

Since errors in this portion of the system are difficult to debug, it is recommended that you determine the method you will use to satisfy the NATPARM requirements before you begin the installation process. Any modifications to the Natural dynamic parameters must be verified in the interactive mode.

TP Monitor and Natural Nucleus Considerations

Con-nect SNADS requires CICS LU6.2 support for the physical transmission of messages and documents between nodes. Additionally, CICS interval control facilities such as the CICS `START` command to schedule the "server programs" are required. Therefore the "server programs" application of Con-nect SNADS must be executed by Natural running under CICS.

Because the "end user" application does not perform the physical transmission of data, you can configure your Con-nect SNADS system so that users are supported from a TP monitor even though the "server programs" application runs from Natural under CICS.

This can be achieved by using a single Con-nect spool file that is shared by the "server programs" application under CICS and the "end user" application. The "end user" application can run in another CICS or a different system such as Com-plete.

For this configuration alternative, the following are required:

- Two Natural nuclei, one for the CICS environment and the other for the "end-user" environment:
 - The CICS Natural nucleus must satisfy all installation requirements for the "server programs" application.
 - The Natural nucleus on the "end user TP monitor" must satisfy all installation requirements for the "end user" application.
- Optionally, a single Natural system file (FNAT) can be shared between the nuclei.

In addition, the following requirements must be satisfied for the Con-nect system files and spool (telex) file:

- A single Con-nect "spool" (telex) file must be shared between the Natural nuclei (logical file number 223 in the NATPARM) for the "end user" and "server programs" applications.
- A link between the Con-nect system file(s) and the shared Con-nect spool file must be created. The following is a description of how this link is created:
 - For each Con-nect system file you have a "home node record" must be defined to the spool file using the Initialize Node File function on the Con-nect "Administration - External Mail Nodes" screen. For further information on the Initialize Node File function, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance*, sub-section *Define Local Node*. This process creates a record with a name (used as the DGN for SNADS), the current physical DBID, and the current physical FNR of the logical file 251.
 - If separate logical 251 files are used, you must invoke Natural with a different logical file 251 setting each time you write these records in order to properly value the home node record. This can be achieved by using the Natural dynamic parameter LFILE.

- If you move a Con-nect system file to a different database or file number, you must re-initialize the home node record with the new setting before mail can be sent or received.
- You can verify the correlation of DSUNs to DGNs and their physical locations (DBID,FNR) by selecting the Control Maintenance function in the Con-nect "SNADS Administration Menu" screen. For further information about the Control Maintenance function, refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS*, sub-section *Control Maintenance*.

Con-nect SNADS and Natural Security

Natural assigns a user ID to both non-interactive and interactive sessions. By default, Natural passes the current user ID to the Adabas nucleus. The user ID allows Adabas to access any end of transaction (ET) data that can be stored for that particular user ID. However, Adabas cannot accept simultaneous sessions with the same user ID. This restriction can cause problems for the DS_RECEIVE, DS_ROUTER_DIRECTOR, and DS_SEND programs. If Natural Security is implemented, more than one instance of these programs can be active simultaneously and all of the programs may attempt to logon with the same Natural Security user ID.

Therefore, if Natural Security is installed in the environment in which Con-nect SNADS is to be implemented, user IDs must be reserved for the Con-nect SNADS "server programs". There are two methods that can be used to avoid conflicts resulting from more than one active session assigned to the same user ID.

The first and most convenient method is to prevent Natural from passing user IDs to Adabas. This is accomplished by assigning a value of blank to the Natural keyword ETID, i.e. ETID=' ' (blank). Because the Con-nect SNADS "server programs" application does not use the ET data facility, data integrity problems will not occur. In this case, only one user ID is necessary for the "server programs" application.

Note:

The ETID setting can be modified in the Natural Security definition or in the dynamic parameter settings in CONSNADS.

However, when ETID is set to blank, Natural does not issue OP and CL commands at the beginning or end of a session. Nevertheless, the module CSDBAS can be linked to the front- and back-end programs to issue OP and CL commands against all databases which are specified in the CONSNADS parameter module.

The second and less convenient method is to allow the appropriate Con-nect SNADS server programs to manage a pool of user IDs. With this method, a user ID which is currently inactive is selected and assigned to a particular function. User IDs consist of a one-to-six character prefix and a two-digit number from 01 through 99 and are specified in the CONSNADS parameter module during the installation process. The two-digit number represents the number of user IDs and should reflect the expected amount of parallel activity.

For example, if you specify XYZ as the prefix and 02 for the number of user IDs, the "server programs" will use either XYZ01 or XYZ02 as a user ID. All user IDs must be defined to the Natural Security system. Also, a password must be associated with each user ID; the same password can be assigned to all user IDs.

This method is not recommended for use. It is supported for compatibility reasons only and may be withdrawn in future versions.