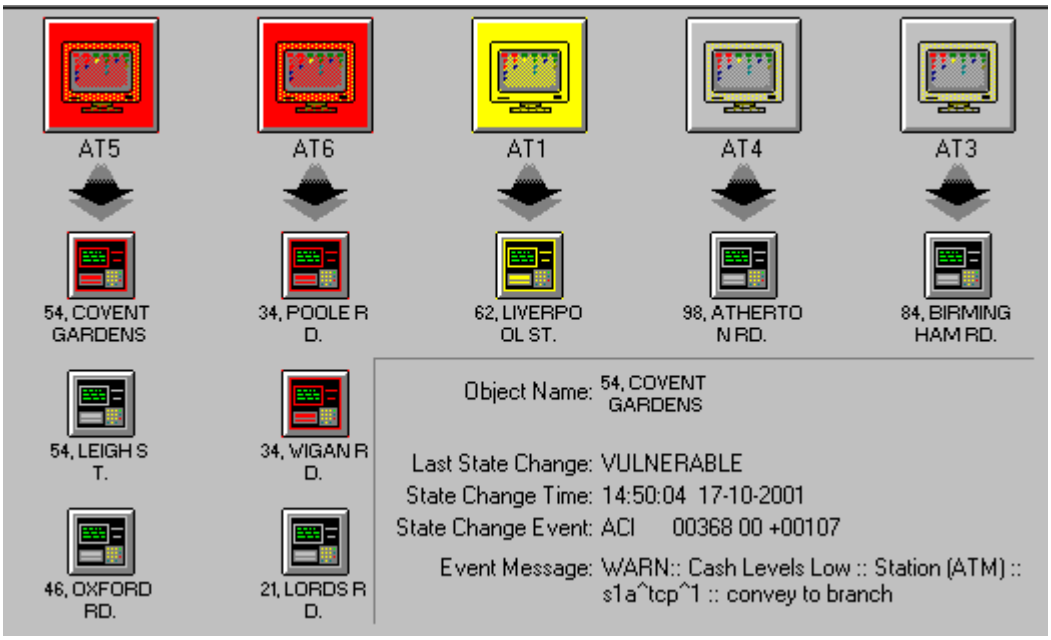
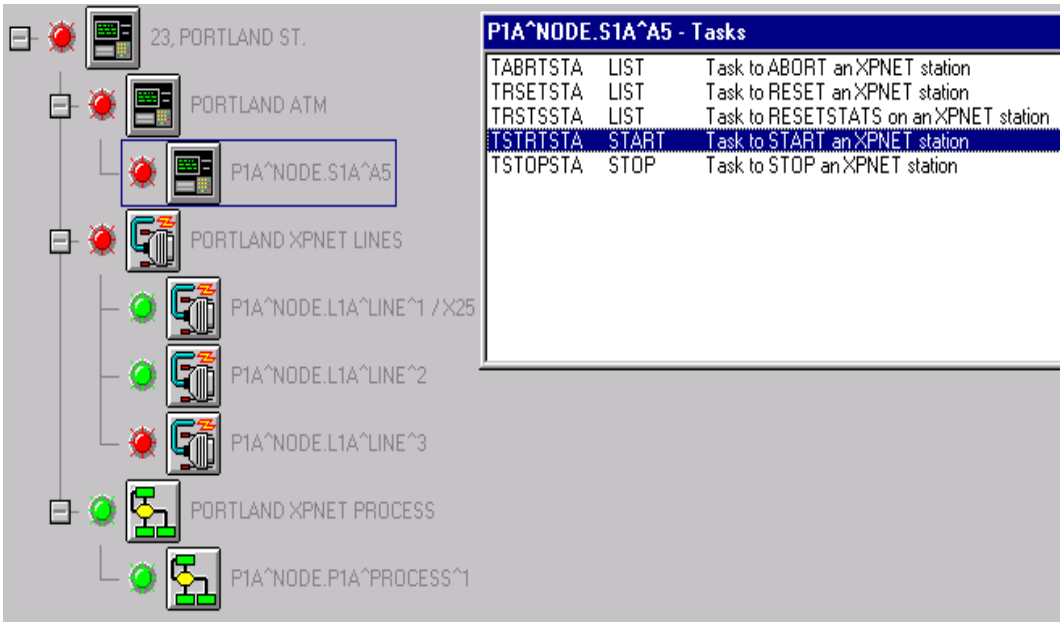


Using Reflex ONE24
To Monitor BASE24



Document Ref: Reflex / Base24
Issue / version: 1.0
Status: Definitive
Number of pages: 32 (including this page)

Date: 070609
Author: ITL - MAW

Contents

1. Introduction

1.1. Overview

This document details the best approach to using the BASE24/XPNET agents available within the Reflex product.

Reflex ONE24 manages single and multiple instances of the BASE24 application across an entire HP NonStop network, offering tight integration within an all-encompassing graphical interface.

An XPNET automatic detection utility is available to extract a copy of the various XPNET nodes, stations, lines, links, processes and device entities using the NCP (network control point) servers of BASE24. This detection will populate the Reflex database with the required records to enable graphical component trees to be built up using the Reflex GUI interface. These object trees can be tailored in any way depending upon how users wish to see the various XPNET relationships. For instance, an object tree can be built to group a station with related lines and processes. The detection utility can also produce default trees where all like XPNET components are grouped under a single display icon for each XPNET object node, e.g. P1A^NODE STATIONS, P1B^NODE LINES.

Once an XPNET graphical tree has been built, a simple click on an individual component will provide access to information, status and statistics relating to that XPNET object. This is provided using a status agent module available within the Reflex ONE24 product, interacting with the NCP-SERVERS of the BASE24 application. This feature enables users to make use of the integrated network display of Reflex to access XPNET component data rather than using TACL prompts and the NCPCOM conversational interface.

Reflex provides quiet, CPU friendly monitoring of all Tandem Guardian sub-systems and applications. This is achieved by taking advantage of the DSMS sub-system. Reflex can be configured with BASE24 and XPNET EMS exception alerts that will automatically be mapped against the detected XPNET graphical trees of Status Monitor. From here, users can acknowledge events and invoke XPNET tasks to remedy problems. A default list of tasks is available in Reflex ONE24, by way of a transit file that can be imported into the Reflex database using a standard deliverable utility.

A list of received EMS events can be displayed for each XPNET entity providing a comprehensive history of issues and states over hours, days and weeks.

The rest of this document verifies and describes the correct set-up of your Reflex application to make use of the BASE24 monitoring capabilities available within the Reflex ONE24 product.

Reflex has optional modules for:

Comprehensive file monitoring and checking file arrivals
Key, busy and looping process monitoring
X25 monitoring and ASAP data analysis
Spooler monitoring
CPU and disk monitoring
Text to EMS conversion
Enterprise manager delivery of exception alerts
Mobile SMS paging
Task Automation
Component rules analysis for SLA service monitoring
Comprehensive EMS console event viewing and filtering

These features are outside the scope of this document. For an overview of the Reflex 80:20 product and Reflex ONE24, visit the ITL WEB site at www.insidertech.co.uk.

2. BASE24 Set-up of Reflex ONE24

2.1. BASE24 XPNET Version

Reflex ONE24 agents have been engineered and tested against version 3.+ of the XPNET layer of the BASE24 application, by staff at Insider Technologies Limited (ITL).

Any questions on the installation of the Reflex ONE24 product or the features currently available should be directed at ITL. For the latest information on features currently available within Reflex 80:20 or Reflex ONE24, check out the ITL WEB site:

www.insidertech.co.uk or email support@insidertech.co.uk

2.2. The Reflex ONE24 XPNETCNF Configuration File

This edit file contains various configuration values to be used by both the XPNET automatic detection utility 'XPNETADC' and the Reflex ONE24 status agent server ACI-XPNET-AGENT (XPNETAGC). This file will need to be modified as part of Reflex ONE24 BASE24 monitoring.

This edit file is always contained within the Reflex data files sub-volume, e.g.

\INSIDER.$DATA02.RFLXDAT.**xpnetcnf**

NOTE: If you have received an XPNETCNF file as part of a Reflex ONE24 BASE24 delivery, place a copy of this in your Reflex data files sub-volume, e.g. RFLXDAT. Check to see if you already have a copy of this file.

The configuration values are listed below in the their respective order within the edit file.

1. **User** – represents the user portion of the key into the XPNET security system. If the security is turned off for the XPNET layer of BASE24 then this value does not need to be supplied. If security is turned on, then an appropriate BASE24 user needs to be supplied here, e.g. *BASE24/USER*. This is currently the default. See section 2.3.
2. **Sess_id** – identifies an end user session in cases where multiple openers may be logged on to the same user definition. This can be any 16-character printable ASCII value. The sess-id is logged in the command audit event. The default is '*rflx24cmndissued*'.
3. **User_info** – represents information needed for user verification. This must match the password entered into the NCSS record for the user-id/node combination. This is only true if the security is turned on the XPNET layer. See section 2.3.
4. **Ncp_val_abnormal** – one of the states of an XPNET component. The default is down (DN). This will show as red in Reflex ONE24 Status Monitor if requesting exact XPNET status using the floating toolbar. If wanting to show this as green (UP) or blue for vulnerable (VU), change value appropriately. This is also true of items 5 to 10 below.
5. **Ncp_val_configured** – the default is vulnerable (VU).
6. **Ncp_val_started** – the default is up (UP).
7. **Ncp_val_starting** – the default is vulnerable (VU).
8. **Ncp_val_stopped** – the default is stopped (DN).
9. **Ncp_val_stopping** – the default is vulnerable (VU).
10. **Ncp_val_suspended** – the default is vulnerable (VU).
11. **Ncp_server** – this is the name of the BASE24/XPNET server which is usually 'SERVER-NCP' within the BASE24 Pathway. This is the default.
12. **Ncp_process** – this should be set to the process name of the production / live Pathway for the BASE24 application, e.g. $PPMN. This is the default.

2.3. A Read Only BASE24 User

If the security for the XPNET layer of the BASE24 application is turned to 'ON' then parameters [1] and [3] above will need to be supplied. This XPNET parameter setting can be verified by accessing the PATHCOM interface for your elected BASE24 application and executing an info request on server 'SERVER-NCPI'. The parameter 'ENABLE-SECURITY' within this server configuration will detail whether or not the security is switched on or off.

The ENABLE-SECURITY parameter is used to enable or disable security checks within SERVER-NCP. If ENABLE-SECURITY is set to ON, the NCSP and NCSS files of BASE24 are used to check a user's access to a node and command. Security violations result in the generation of a security violation event containing user and session information. If ENABLE-SECURITY is set to OFF, the NCSP and NCSS files are not checked. The default value for this parameter is ON.

If switched off then parameter [1] and [3] above need not be supplied and can be set to spaces or left at the default values. If switched ON then a special Reflex ONE24 BASE24 user needs to be set-up (or a current low security BASE24 user utilised) within the BASE24 application to allow read only access.

This user requires only the ability to see the BASE24 XPNET object components, i.e. XPNET nodes, stations, lines, links, processes and devices. This ensures that the automatic detection utility and status agent can issue the following commands programmatically through the NCP servers:

- Listobjects - automatic detection - TACL
- Status - status agent - GUI
- Info - status agent - GUI
- Statistics - status agent - GUI

This is the same user and password when using the NCPCOM conversational interface when security is set to on. Update parameter [1] and [3] in the *XPNETCNF* file discussed in section 2.2.

Remember that the password and user are case sensitive and that the XPNETCNF file needs to reflect the user set-up exactly.

2.4. XPNET Object Types and Sub-types

Every object or entity with the Reflex application has an associated type and subtype. Here are some examples:

Tandem Component	Type	Subtype
CPU	TANDEM_HARDWARE	CPU
DISK	TANDEM_HARDWARE	DISK
SERVER	PATHWAY	SERVER
PROCESS	TANDEM	PROCESS

The reason behind this type association is so that tasks, status agents and Status Monitor graphical icons can be linked with component types, such as the above. In the case of tasks, only pertinent tasks are listed in the Status Monitor window for particular component types. Some types and subtypes have a corresponding status agent. An agent can probe a component sub-system for more detailed information on that object. The component information can then be displayed in a pop up window in the Status Monitor screen.

The types and subtypes for the BASE24 XPNET components are as follows:

XPNET Component	Type	Subtype	Status Agent
NODE	BASE24_XPNET	NODE	ACI-XPNET-AGENT
STATION	BASE24_XPNET	STATION	ACI-XPNET-AGENT
LINE	BASE24_XPNET	LINE	ACI-XPNET-AGENT
LINK	BASE24_XPNET	LINK	ACI-XPNET-AGENT
PROCESS	BASE24_XPNET	PROCESS	ACI-XPNET-AGENT
DEVICE	BASE24_XPNET	DEVICE	ACI-XPNET-AGENT

These need to be added to the list of types and subtypes delivered as standard with the Reflex product. To do this, carry out the following (*see note below*):

1. Logon to the Reflex application with an appropriate Guardian user who has been given set-up rights.
2. Click on the 'Monitor' option above the main toolbar, which is the sixth option along above the Gateway graphic.
3. Click on the 'Network Set-up' option to display the Overdrive Configuration Window. Maximise this window.
4. Subsequently click on the 'Type Set-up' tab to show a list of pre-delivered types and subtypes.
5. Scroll down the list to look for the above types and subtypes for the XPNET components. If they are not present then type them in *exactly* as specified above and click on the add button '**+**' on the window toolbar. The list of icons, which correspond to the types and subtypes, are contained in the back of the Reflex 80:20 User Manual.

NOTE: if you have received an SQLINUP file as part of a Reflex ONE24 BASE24 delivery, the above steps are not required. Carry out the following procedure:

1. Logon to the Tandem as the Guardian User who owns the Reflex application.
2. Volume to the object sub-volume, e.g. RFLXOBJ.
3. Run the RSQLDEFS file in this sub-volume to load the Reflex SQL defines.
4. Access the SQLCI interface by typing 'SQLCI' at a TACL prompt.
5. Obey the SQLINUP file supplied to add the XPNET component types and subtypes
6. Exit from the SQLCI interface. The Types and Subtypes can then be viewed by carrying out steps 1 to 4 above.

2.5. Checking the Reflex ONE24 PATHWAY Configuration

Reflex ONE24 requires a PATHWAY server entry to be added to the PWCONF file contained in the object sub-volume of Reflex e.g. RFLXOBJ. This contains an entry for each of the Pathway servers in the Reflex application. The Pathway server description is as follows:

```
RESET SERVER
SET SERVER                  IN rflxdat.DATACONF
SET SERVER                  LINKDEPTH 1
SET SERVER                  MAXLINKS 5
SET SERVER                  MAXSERVERS 1
SET SERVER                  NUMSTATIC 0
[>>>>>>>>>>>>>>>>>>         Site Specific Data Reference  START
SET SERVER                  PRI 150
SET SERVER                  CPUS 1:0
SET SERVER                  PROCESS $RXXP
SET SERVER                  PROGRAM rflxobj.XPNETAGC
[>>>>>>>>>>>>>>>>>>         Site Specific Data Reference  END
SET SERVER STARTUP "XA $XPNET XB SERVER-NCP XC 1500 XD 0 XE 0 XF 0 "
SET SERVER                  TIMEOUT 10 MINS
SET SERVER                  TMF ON
SET SERVER                  HIGHPIN ON
ADD SERVER                  ACI-XPNET-AGENT
```

NOTE: If you have received a PWCONFUP file as part of a Reflex ONE24 BASE24 delivery, add the contents of this file to your PWCONF file if not already present. Change the 'IN' and 'PROGRAM' references to point to your own sub-volume locations.

It is important to set the parameters on the STARTUP line to their appropriate values. Only the 'XA' and 'XB' parameters need to be reconciled as the XC to XF parameters are not used. *These should be left at their supplied defaults*.

The 'XA' parameter represents the live production BASE24 environment Pathway, e.g. $PPMN. This should be set to the same value as '**Ncp_process'** as discussed in section 2.2 of this document.

The 'XB' parameter represents the NCP server within the BASE24 application Pathway. The default is SERVER-NCP. It is unlikely that this will differ on your site.

2.6. Checking REFLEX ONE24 XPNET Executables

Reflex ONE24 requires two object files to be placed in the object sub-volume before BASE24 XPNET monitoring can proceed. These are as follows:

- XPNETADC - XPNET menu driven detection utility
- XPNETAGC - XPNET status agent to supply info, status and statistics

The XPNETADC executable can be run from a TACL prompt (see section 3) to provide for automatic detection of XPNET components. It does not require a Pathway server to be set-up and is a menu driven utility. The second object, XPNETAGC requires a Pathway server as discussed in section 2.5, and allows detailed information to be passed back to the Reflex Status Monitor graphical window on user request.

NOTE: If you have received both XPNETADC and XPNETAGC object files as part of a Reflex ONE24 BASE24 delivery, place them in the Reflex object sub-volume. Both files interact with the NCP servers of XPNET but the XPNETADC detection executable also populates the Reflex SQL database. For this reason it will need SQL compiling. If required, carry out the following steps:

1. If not already, logon to the Tandem as the Guardian user who owns the Reflex application.
2. Volume to the Reflex object sub-volume, e.g. RFLXOBJ.
3. Run the RSQLDEFS file in this sub-volume to load the Reflex SQL defines.
4. Type the following:

   >SQLCOMP / in XPNETADC, out $s.#xprflx, pri 50, name $rfsqc/ catalog =reflex_catalog
5. Check PERUSE to ensure the SQL compilation was successful.

Once having applied any of the amendments referred to in this section of the document, stop and start the Reflex application to pick up the ONE24 modifications. To do this, carry out the following steps:

1. If not already, logon to the Tandem as the Guardian user who owns the Reflex application.
2. Exit any Reflex GUIs currently running.
3. Volume to the Reflex object sub-volume.
4. Obey the STOPRFLX file in this sub-volume.
5. Once stopped, obey the RUNRFLX file in the same sub-volume.
6. Re-start and logon to the Reflex GUI.

3. Auto-Detection of BASE24/XPNET

3.1. The Reflex ONE24 BASE24 XPNETADC Detection Utility

It is recommended that this section be read in its entirety before deciding on a particular approach to graphically representing your BASE XPNET components.

This utility allows for the detection and extraction of a complete copy of the XPNET components that make up a given XPNET node. These components are subsequently inserted into the Reflex SQL database to provide easy configuration of the product for BASE24 XPNET monitoring. To run the utility, carry out the following steps:

1. Logon to the Tandem as the Guardian user who owns the Reflex Pathway application.
2. Volume to the Reflex object sub-volume, e.g. RFLXOBJ.
3. Type: >RUN XPNETADC / in RFLXDAT.XPNETCNF / [**return**]
4. The following menu will appear:

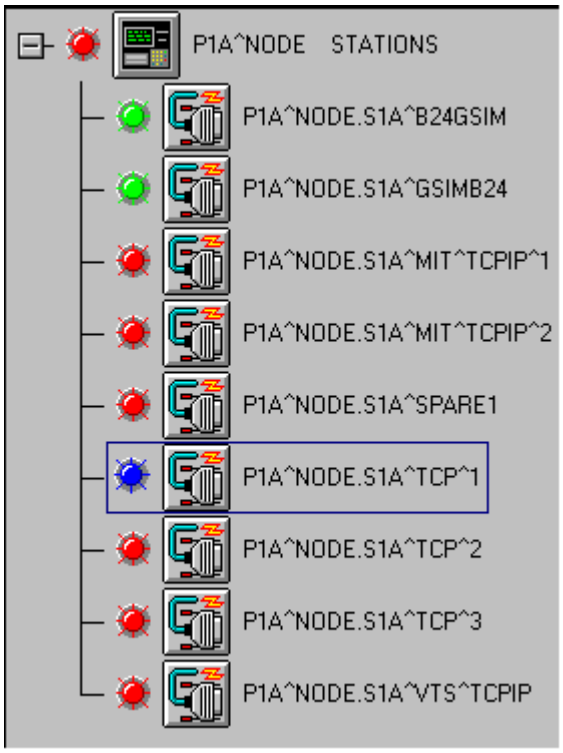
```
Node : \INSIDER        Reflex BASE24 XPNET Autodetect Menu


  [ 1] Auto-Detect NODES          [ 7] Delete NODES
  [ 2] Auto-Detect LINES          [ 8] Delete LINES
  [ 3] Auto-Detect STATIONS       [ 9] Delete STATIONS
  [ 4] Auto-Detect PROCESSES      [10] Delete PROCESSES
  [ 5] Auto-Detect LINKS          [11] Delete LINKS
  [ 6] Auto-Detect DEVICES        [12] Delete DEVICES


  [13] Enter a Preferred Reflex CLASS for SMON Display
  [14] Object Tree/Object Detect - Default Object Tree
  [15] Autodetect Objects Outside of SMON Warm-Boot


   0. Exit Menu


Option >
```

This menu allows for both the detection and deletion of XPNET components from the Reflex database. The *default* detection will insert all like components under a single Status Monitor group. This will be placed below the BASE24 class heading (B24) for a given BASE24 XPNET node. This can be seen clearly in the following bitmap.

If option [3] on the menu 'Auto-Detect STATIONS', is selected, the user is prompted to enter the XPNET node to probe for station components.

```
Option > 3


Enter XPNET node for stations to be autodetected, e.g. P1A^NODE:
XPNET Node> P1A^NODE
```

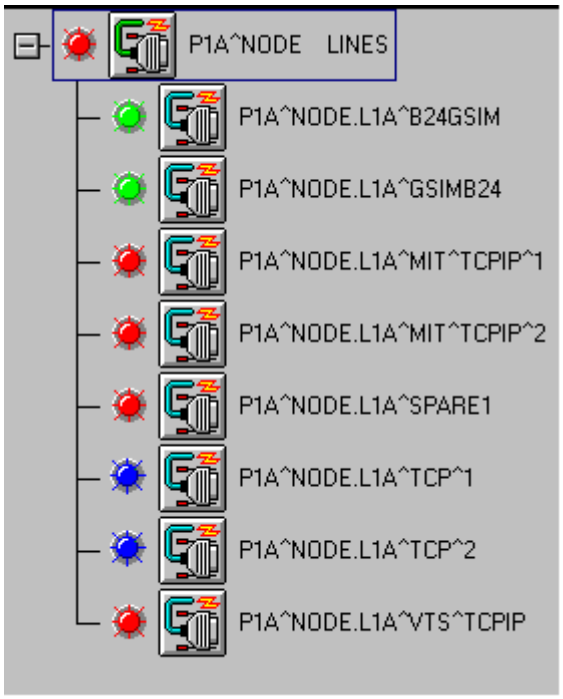
This will result in a group under the B24 class heading, called 'P1A^NODE STATIONS'. Beneath this will be placed all the stations belonging to that XPNET node. This is shown in the following bitmap (NOTE: the status of components is discussed in a later section).



Similarly, if option [2] on the menu 'Auto-Detect LINES', is selected, the user is prompted to enter the XPNET node to probe for line components.

```
Option > 2


Enter XPNET node for lines to be autodetected, e.g. P1A^NODE:
XPNET Node> P1A^NODE
```

This will result in a group under the B24 class heading called 'P1A^NODE LINES'. Beneath this will be placed all the lines belonging to that XPNET node. This is shown in the following bitmap (NOTE: the status of components is discussed in a later section).



This is the default automatic detection for all XPNET node components. The detection of the actual XPNET nodes themselves will result in all XPNET nodes for a given Pathway being placed under a top-level group called 'XPNET NODES'.

If running the XPNETADC ONE24 detection utility, the default discovery will result in all like components being placed under a group name with the XPNET node name followed by the type of component as shown above. This can be changed to provide a more tailored aesthetic look for your BASE24 XPNET layer and this discussed in the next section.

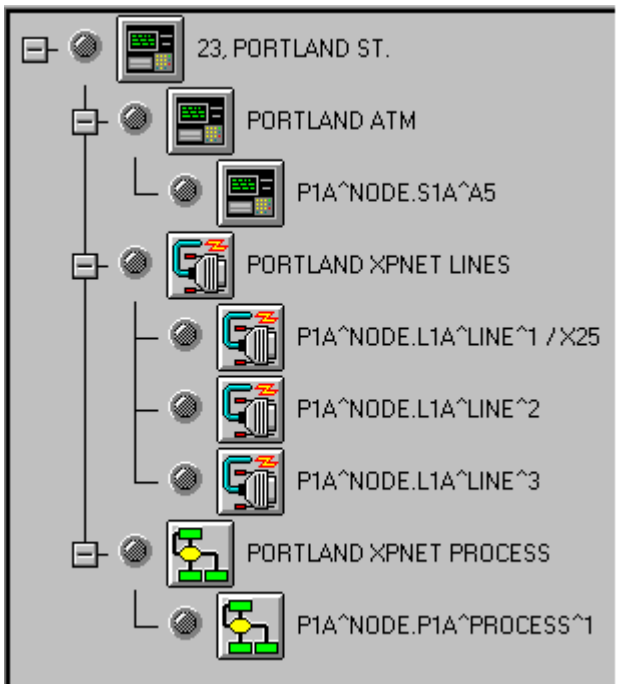
If more XPNET components are configured in the XPNET layer of BASE24, the XPNETADC detection utility can be used to redetect and so rebuild the object trees of Reflex Status Monitor. Before this can be done, a 'DELETE' should be carried out for the appropriate object tree (see NOTE below before doing this). In the case of the last example, 'P1A^NODE LINES', this is carried out by selecting option [8] from the menu and typing the appropriate XPNET node.

```
Option > 8


Enter XPNET node for lines to be deleted, e.g. P1A^NODE:
XPNET Node> P1A^NODE
```

IMPORTANT: Detecting all of your XPNET components such that they are grouped under a main heading group name, that is 'XPNET node name' followed by type, e.g. P1A^NODE STATIONS, then performing a deletion followed by redetection can be carried out as often as required. This *may not* be the case if using a less grouped approach for displaying XPNET entities; e.g. every station (ATM or Point of Sale Terminal) has its own presence in a top-level Status Monitor matrix (see ATM diagram below). See next section for tailoring different graphical layouts for your XPNET components and approaches to adding to Reflex ONE24, newly configured BASE24 XPNET components.

3.2. Options Available for Tailoring Graphical Displays

Different approaches for displaying the XPNET layer of BASE24 can be taken in Reflex ONE24 other than the default displays talked about so far. This can be achieved by using some of the options available in the XPNETADC ONE24 detection utility but also making use of the Status Monitor set-up screens. These allow for the dragging and dropping of objects into component trees for a greater separation of business service areas or geographical locations. In the latter case, ATMs (stations) as an example, can be separated into regions, streets or most and least sensitive in terms of customer usage. Lines associated with stations can be grouped under the same heading so that relationships can be maintained graphically making troubleshooting easier. This is especially true when using tasks, see section 6.



What follows is a discussion on the ways that Reflex ONE24 can be used to produce the best graphical representation of the BASE24 XPNET layer for your particular requirements. Later options may provide a more appropriate strategy for your particular on-site monitoring needs.

Option [13] – Enter a preferred Reflex ONE24 class for SMON display

The default for any detected XPNET components is to place the resulting group of components under the 'B24' class as shown in section 3.1. Option [13] allows the user to change the class under which detected groups are stored. This can also be a totally new class name not currently seen in the Status Monitor display, e.g. NEW, ACI or XPN.

To place components belonging to say 'P1A^NODE' XPNET node under a P1A class, request option [13] from the XPNETADC detection menu and type ***P1A*** (or a preferred class) at the prompt 'CLASS Node>'.

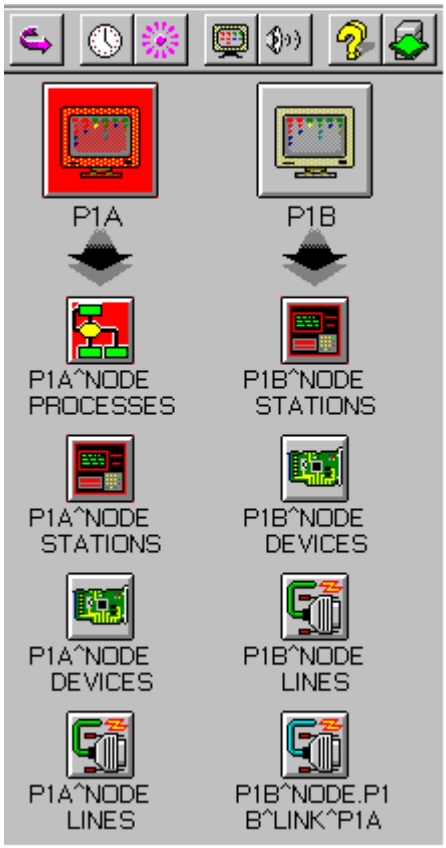
```
Option > 13

Enter a Reflex CLASS name under which to store Autodetection:
CLASS Node>
```

Using options 2 to 6 on the XPNETADC detection menu, build the Reflex ONE24 Status Monitor groups for the relevant *P1A^NODE* component types (or the XPNET node you are detecting for).

NOTE: The XPNET nodes themselves are detected and inserted under a single group and so should be placed under a generic class if using this class naming approach, e.g. XPN or B24 could be used for XPNET nodes.

If you have a second node to detect, e.g. P1B^NODE, enter option [13] once again from the XPNETADC detection menu and type 'P1B'.

Using options 2 to 6 on the XPNETADC detection menu, build the Reflex ONE24 Status Monitor groups for the relevant *P1B^NODE* component types (or the XPNET node you are detecting for). The resulting display using this approach to XPNET detection will look as follows (the status is discussed in a later section):



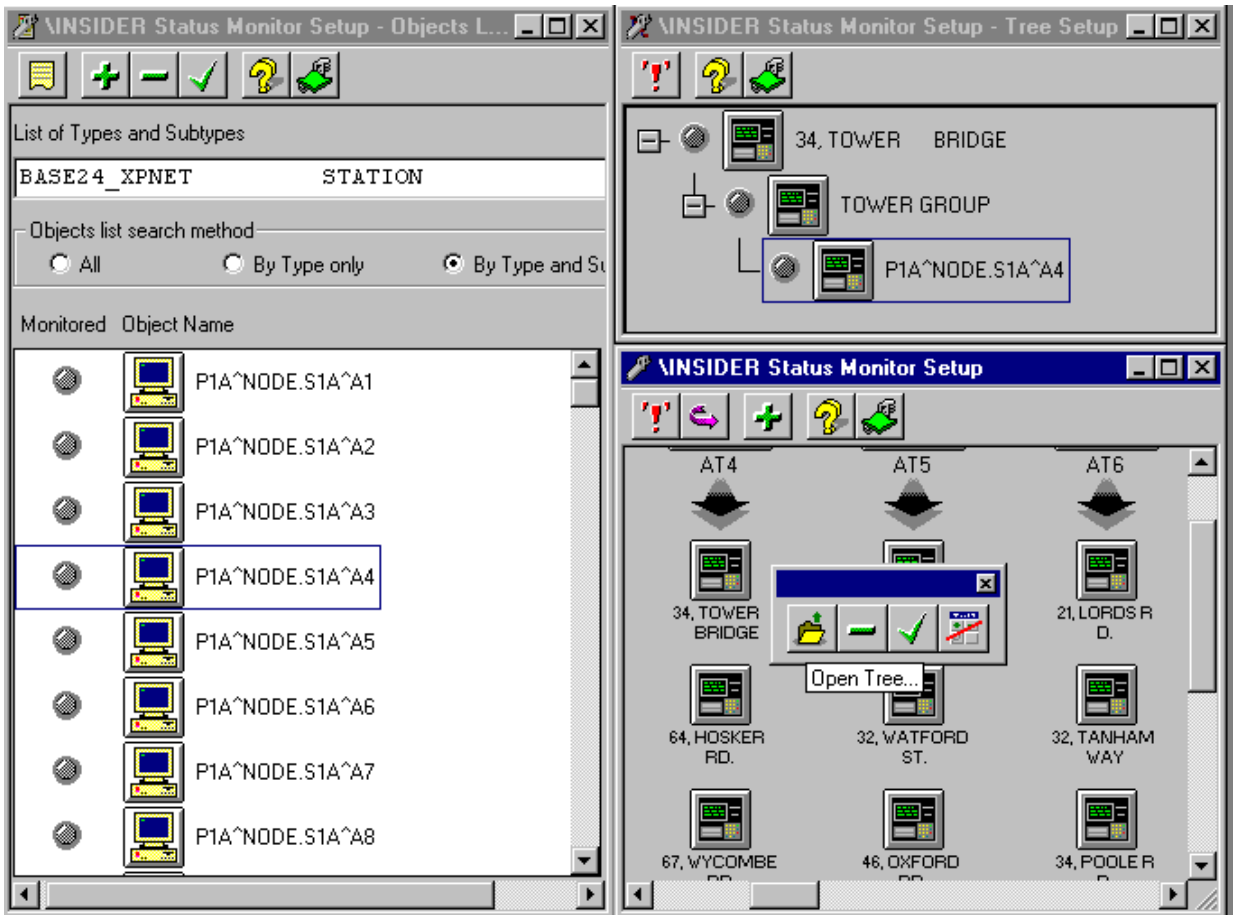
Option [14] – Object Tree / Object Detect – Default Object Tree

Rather than building XPNET component trees using the detection menu, option [14] allows for the detection of just the XPNET objects themselves. These are inserted into the object SQL table, e.g. RFLXDAT.OBJECTQ. From here they can be dragged and dropped using a Reflex GUI window, into groups that you have named yourself.

```
Option > 14

Enter 1 - Object Tree / 2 - Object Level Auto-Detection:
1 or 2>
```

Enter 14 at the XPNETADC menu as shown above and then enter 2 to signify that you just require the raw XPNET components to be detected rather than the tree building to be performed. After this toggle has been applied, any detection that takes place using the options 1 to 6 on the detection menu will only detect the XPNET objects and *not build* the Status Monitor trees.

After detecting the XPNET objects, the user can access the object list window of Reflex and drag and drop objects into appropriately named groups. This is shown in the following bitmap:



Outlined below are the steps used to create a group called '34, Tower Bridge' and drag an appropriate station (ATM) under it for monitoring.

1. Logon to the Tandem with the Guardian user who owns the Reflex Pathway.
2. Volume to the Reflex object sub-volume, e.g. RFLXOBJ.
3. Run the XPNETADC ONE24 detection utility as follows (RFLXDAT is the location of your Reflex data files):

   > RUN XPNETADC / IN RFLXDAT.XPNETCNF / **[return]**
4. Enter option [14] and set detection type to '2' – Object Level Auto-Detection.
5. Enter option [3] to detect XPNET stations.
6. Enter the XPNET node you wish to probe and extract a copy of the station components to insert into the Reflex database, e.g. P1A^NODE.
7. Enter '0' to exit from the XPNETADC detection menu.
8. Logon to the Reflex ONE24 GUI with a user who has access to Reflex set-up facilities, e.g. a Reflex ONE24 Administrator user.
9. Click on the 'OD set-up' icon on the main toolbar.

10. Maximise the resulting window.

11. Click on either an existing class icon, e.g. ATM, B24, BAT or click on '**+**' icon on the toolbar to enter a new class name, e.g. AT1, XPN, ACI. Enter the group name '34, Tower Bridge' (or your own preferred name) and choose an icon for display, e.g. ATM. A list of the icons associated with the 3 letter acronyms is listed in the Reflex 80:20 User Manual at the back.

12. Click on the '**+**' icon on the set-up window toolbar and acknowledge the message for confirmation of the added group/class function. The screen will be refreshed and your top-level group will be displayed appropriately under the chosen class.

NOTE: In some cases, the text may be wrapped to the next line of the display for a group name breaking a word into 2 fragments. If this is not preferred, delete the group by left clicking on it and re-add but placing some spaces between it and the next word, e.g. '34, Tower Bridge'. This will force the set-up code to throw the next word to the next line rather than breaking it over 2 lines. NOTE: Reflex allows up to 11 characters per line.

13. Left click on the group you added and click on the folder icon on the floating toolbar to drill down.

14. Click on the 'Window' option above the main toolbar of Reflex and tile the display vertically. Three windows will appear. The top-level display you have just come from, the object list window and the next level set-up window.

15. Minimise the top-level window and re-tile vertically so that only the object list window and next level set-up window appears.

16. Right click on the group name icon you added and add another group name. The reason this is done is so that in your eventual display, your group name is not replaced by the object name you will eventually add since there is only one object in the group.

17. Type the name of a group and click on the '**+**' button to insert it.

18. On the object list window, click the pager button for listing 'By Type and Subtype'.

19. Click on the 'down arrow' under the 'List of Types and Subtypes' heading to show the current Reflex ONE24 types and sub-types.

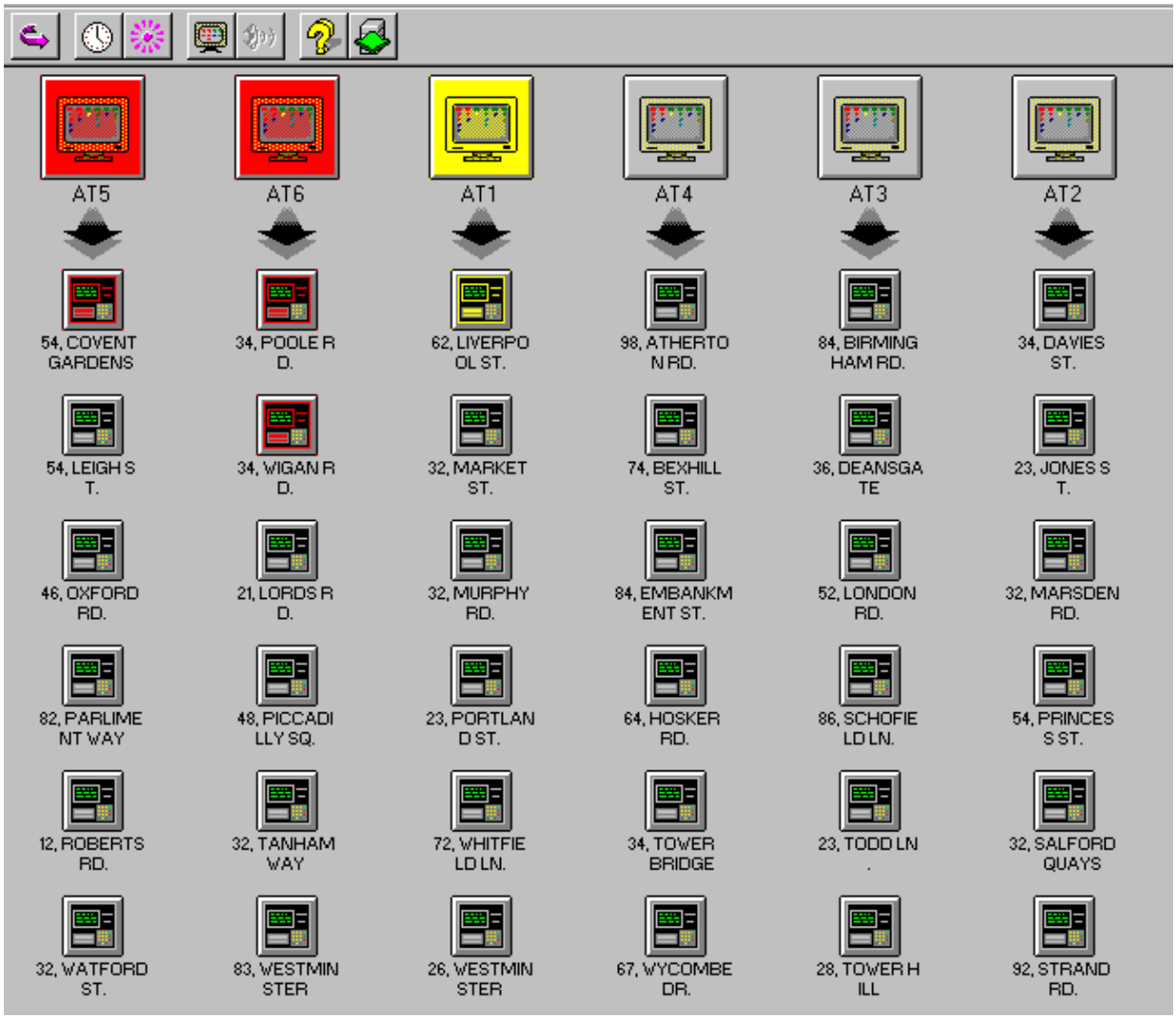
20. Click on the 'BASE24_XPNET STATION' type and subtype and subsequently list the objects using the note paper icon in the top left of the object list window.

21. The stations will be listed. Click and hold on the appropriate station and drag and drop it over the second group you added in next level window display.

22. Finally, click on the '**!**' icon on the next level window to warm-boot the set-up data to show your newly added group containing your elected station in the live Overdrive monitoring window. Click on the confirmation messages for the warm-boot and subsequently click on the 'Overdrive' icon on the main toolbar to see the result.

Repeatedly building up groups in this way allows the user to produce any desired graphical display for XPNET nodes, stations, lines, links, processes and devices. For an ATM network, repeatedly carrying out the above steps after detecting the stations, can lead to an ATM matrix similar to the one displayed below. In larger networks, ATMs can be grouped under a single group. For instance, Oxford Street in London may be home to 40 ATMs that could be dragged and dropped under an Oxford Street top-level group. Each of the stations underneath the group can then be given an alias name for the street number or sequence number.

The OD set-up windows are discussed in more detail in the Reflex 80:20 User Manual.

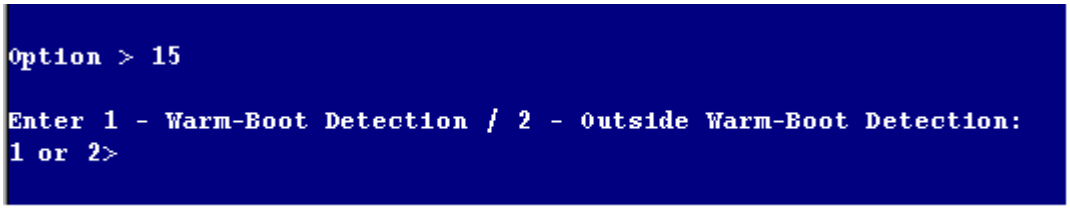


IMPORTANT: If tailoring your own screens without using the XPNETADC ONE24 defaults of placing components all under a group then the deletion options of the XPNETADC menu should NOT be used. Instead, components should be added manually using the Reflex ONE24 GUI. The detection facility is for bulk detection to speed up the set-up process but small modifications to the screen or new XPNET additions should be made manually.

If using the defaults of group detection under XPNET node icons, the deletion options can be used as often as required to refresh those groups.

Option [15] – Auto-detect Objects Outside of SMON Warm-Boot

This option on the XPNETADC detection menu allows for the production of a separate XPNET database. This falls outside of the warm-boot mechanism. The sum result of this mode of detection is that the resulting trees cannot be accessed using the set-up GUIs of Reflex. They can only be seen in the live Overdrive window. Choosing '2' from this option allows no user access to detected XPNET trees.



In large XPNET systems, this will lead to an improved warm-boot response time. The XPNET component trees are always available in the live Overdrive window with any red/blue status logged against individual XPNET components being retained. *This option will be expanded in the next release of the software.*

3.3. Adding XPNET Components Manually After Bulk Detection

It is important that after bulk detection of XPNET components that newly configured stations, lines etc., are added manually to the Reflex database. This is only true if *not using* the default detection options offered for grouping objects and opting for a more tailored Status Monitor screen.

Outlined below are the steps in adding in XPNET components manually using the GUI set-up screen:

1. Logon to the Reflex GUI with a user who has access to Reflex set-up facilities, e.g. a Reflex ONE24 Administrator user.
2. Click on the 'OD set-up' icon on the main toolbar.
3. Maximise the resulting window.
4. If no appropriate XPNET groups currently exist below the class names across the top of the screen, click on either an existing class icon, e.g. ATM, B24, BAT or click on '**+**' icon on the toolbar to enter a new class name, e.g. AT1, XPN, ACI. Enter a group name and choose an icon for display, e.g. ATM. A list of the icons associated with the 3 letter acronyms is listed in the Reflex 80:20 User Manual at the back.
5. Click on the '**+**' icon on the set-up window toolbar and acknowledge the message for confirmation of the added group/class function. The screen will be refreshed and your top-level group will be displayed appropriately under the chosen class.

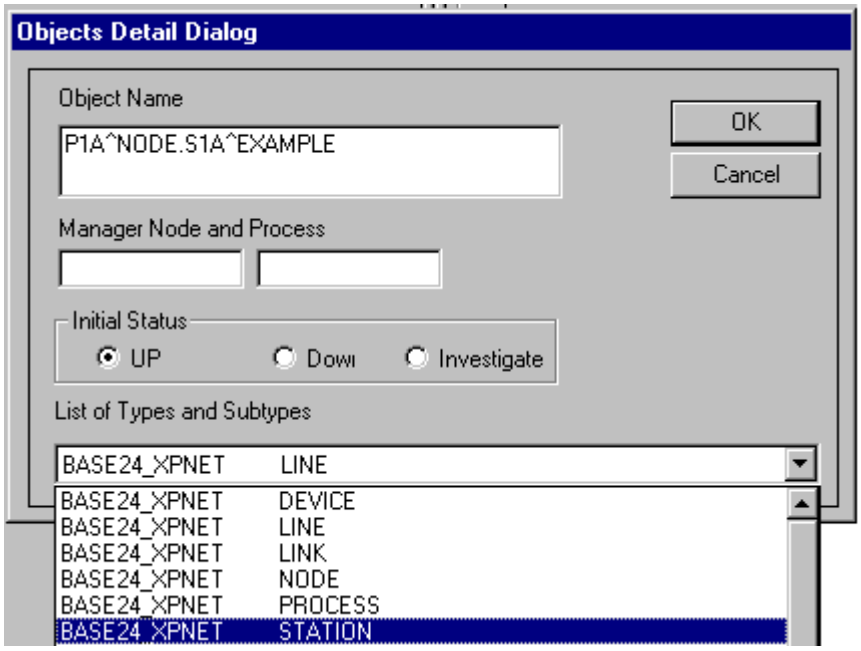
NOTE: In some cases, the text may be wrapped to the next line of the display for a group name breaking a word into 2 fragments. If this is not preferred, delete the group by left clicking on it and re-add but placing some spaces between it and the next word, e.g. '34, Tower Bridge'. This will force the set-up code to throw the next word to the next line rather than breaking it over 2 lines. NOTE: Reflex allows up to 11 characters per line.

6. Left click on the group you added and click on the folder icon on the floating toolbar to drill down.
7. Click on the 'Window' option above the main toolbar of Reflex and tile the display vertically. Three windows will appear. The top-level display you have just come from, the object list window and the next level set-up window.
8. Minimise the top-level window and re-tile vertically so that only the object list window and next level set-up window appears.

9. Right click on the group name icon you added and add another group name. The reason this is done is so that in your eventual display, your group name is not replaced by the object name you will eventually add since there is only one object in the group.

10. Type the name of a group and click on the '**+**' button to insert it.

11. On the objects list window, click on the add button '**+**'. The following window should appear.



12. Add in the name of the XPNET component you wish to add in the format as shown above, e.g. P1B^NODE.S1B^EXAMPLE^STAT, where the component name is prefixed with the XPNET node name.

13. Select the appropriate type and subtype for the newly added XPNET component from the drop down list as shown above.

14. Select from the 'Initial Status' box, whether during a warm-boot, object is to be 'Investigated' (to check status of object) or automatically placed in either an 'UP' or 'DOWN' state, regardless of actual state. EMS events or users requesting status after the warm-boot has occurred will drive the icon status after the warm-boot

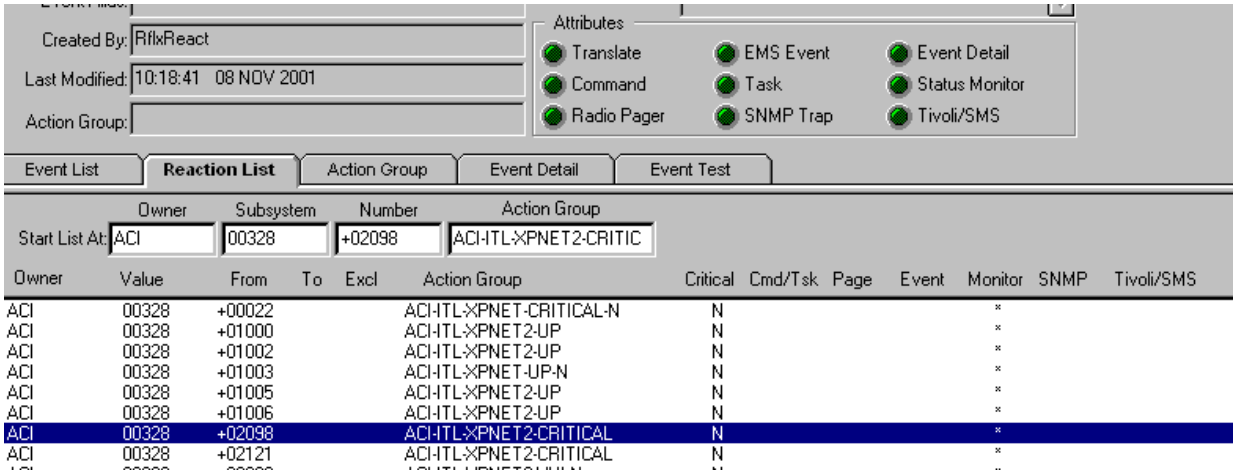
15. Click on the OK button to add the XPNET object and drag and drop it into the group in the next level window. Once the object has been added it can be dragged into any group you choose or can occur in more than one group.

16. Finally, click on the '**!**' icon on the next level window to warm-boot the set-up data to show your newly added group containing your elected station in the live Overdrive monitoring window. Click on the confirmation messages for the warm-boot and subsequently click on the 'Overdrive' icon on the main toolbar to see the result.

4. Configuring BASE24 EMS Events for Reflex ONE24

4.1. The Reflex Reaction Module

The Reflex product allows for the monitoring of chosen EMS event ranges within the BASE24 application. These can be system level problems or issues, such as invalid or unrecognised data passed to certain XPNET processes, right through to ATM (station) specific EMS alerts such as: hopper low, cash out, paper out, etc.

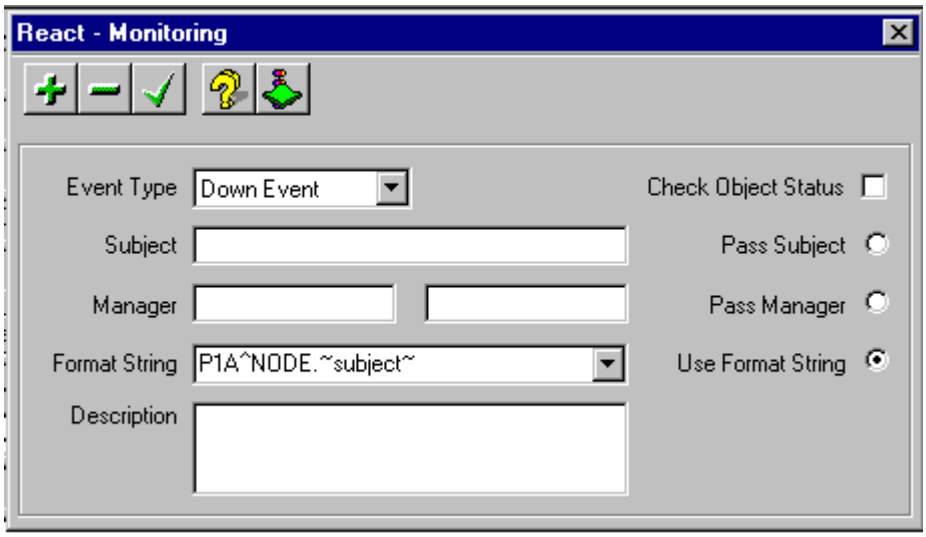


The green light 'Attributes' in the bitmap above represent all the reactions that can be carried out as a result of receiving a BASE24 event. An event relating to any area of the BASE24 application can be mapped either against a generic icon graphic in Status Monitor or an individual XPNET node, station, line, link, process or device.

In order to add an event (or event range), carry out the following steps:

1. Logon to the Reflex GUI with a user who has access to Reflex set-up facilities, e.g. a Reflex ONE24 Administrator user.
2. Click on the Reaction module on the main toolbar of Reflex.
3. Maximise the resulting window.
4. Click on the 'Action Group' tab of the maximised window.
5. Enter the SSID of the event(s) you wish to monitor, e.g. XPNET would be ACI.328.
6. Enter the event or event range of the EMS event(s) you wish to monitor.
7. Enter an Action Group. This is a free text field used to group a reaction(s) that you wish to occur when the given EMS event(s) is passed to Reflex. This Action Group can be used again if just wishing to carry out the same reaction, e.g. toggle the colour of an XPNET component icon from one status to another. Action group 'ACI-ITL-XPNET-CRITICAL' could be used as the action group to toggle the status of a station, line or process to red (critical/down). This action group is subsequently associated with EMS events relating to XPNET components that are in a down state.
8. Enter a desired cover period.

9. Click on the Status Monitor green attribute button that is the middle right reaction. The following window will be shown:



10. Enter the appropriate status you wish to reflect on the graphical Status Monitor screen, i.e. Up, Down or Vulnerable (the latter could be station or line STOPPING for instance).

11. Click on the 'Use Format String' pager button and clear the 'Check Object Status' box. In the latter case, this represents whether Reflex ONE24 function checks the physical or actual status of an XPNET object on receiving an event. This may or may not be appropriate. If an EMS event states that a station has stopped but it has physically started then you may wish to check the object status for this EMS event to be overridden. If the EMS event is informing that 'cash is low' for an ATM (station) then checking the object status will toggle the status of the ATM back to up. In this case we want to display a vulnerable condition even though we are aware of the station being physically up in XPNET. Clearing the object status box in this case, is the preferred setting.

12. Enter the format string as XPNET node suffixed by the subject string in the format shown above. In the case of most XPNET events, the subject of the event is the offending XPNET component. This will be extracted and joined to the XPNET node as entered above and subsequently mapped exactly against one of the graphical icons in the Overdrive window. In the case were the XPNET EMS event has a generic EMS subject token, create an object with a generic name (see section 3.3), e.g. XPNET ISSUES and type 'XPNET ISSUES' into the format string field above. If an event is raised stating a general XPNET issue, this can be mapped directly to the graphical icon 'XPNET ISSUES'.

13. Click on the add button '**+**' on the window to add the record. If you are re-using the same action group in a BASE24 event range, then the details may already exist in which case just click the tick (amend) icon to amend the record. This sets a flag on the action group screen to denote that a reaction has been set-up for the given event or event range.

14. Click on the OK confirmation box and exit the above window.

15. Click on add '**+**' icon on the Reaction window (or amend - tick) to record the addition or modification.

Carry out these steps for other event or event ranges that you wish to monitor. When you have completed the new event additions, carry out the following steps:

1. Click on the '**!**' icon on the reaction toolbar. The following floating window will be displayed.

2. Click on 'Generate Filter Source' button and wait for confirmation of completion. This generates a filter source file to pass your newly added BASE24 events to the Reflex event monitor.
3. Click OK on the confirmation message. Click on the 'Compile Filter Program' and then click OK to the 'compilation started' confirmation message.
4. Logon to the Tandem as the Guardian user who owns the Reflex Pathway application and check PERUSE for successful compilation listings. Alternatively, activate CONSOLE and check for successful compilation EMS messages generated by Reflex.
5. Click on the 'Warm-boot Reaction Servers' to allow the event monitor to pick up the new filter for your newly added event ranges.

See the Reflex 80:20 User Manual for more details about using Reaction module.

4.2. Event Ranges for Multiple XPNET Nodes

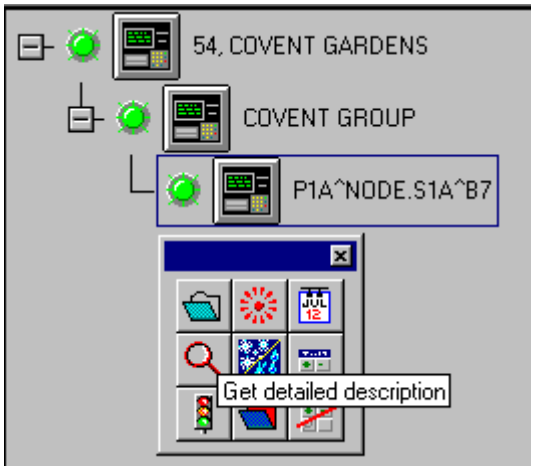
If you have more than one XPNET node within your production / live BASE24 Pathway, then multiple event ranges need to be added to map to the graphical icons of Status Monitor. This is only the case if the reaction is for Status Monitoring.

In the case of the example in section 4.1, add the same event (or event range) with a different Action Group, e.g. ACI-ITL-XPNET-CRITICAL**2**, and assign your second node, e.g. P1B^NODE in the React – monitoring window, e.g. ***P1B^NODE.~subject~***. This will result in 2 different action groups assigned to the same event range. This approach is only necessary for non-generic graphical icons, e.g. P1B^NODE.S1A^STATION, as opposed to 'XPNET ISSUES' as discussed in the previous section.

5. Reflex GUI Display of ONE24 XPNET Information

5.1. Displaying XPNET Information, Status and Statistics

With the inclusion of the Reflex ONE24 status agent in the Reflex Pathway configuration (ACI-XPNET-AGENT), users are now free to click on any of the BASE24 XPNET components for detail. The detail window that is brought up will contain detailed configuration information, status details and statistics for the appropriate object. In order to retrieve detailed information, left click on any XPNET component and the floating toolbar shown in the first bitmap below, will be displayed.



Click on the magnifying glass icon (centre left) to retrieve detailed information in the format shown below for the appropriate XPNET entity:

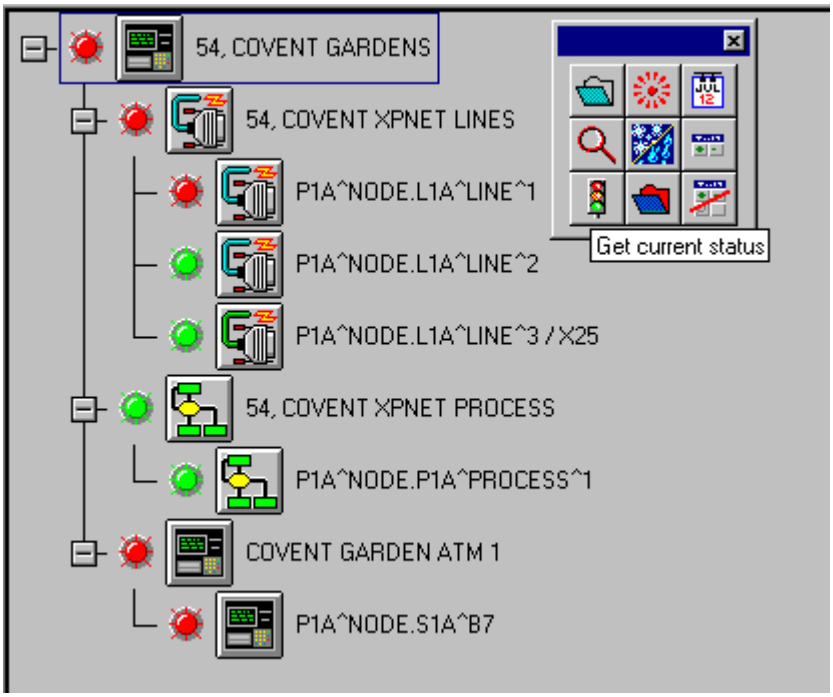
```
TRIBUTARY OF
SYMBOLIC NAME                S1A^B24GSIM
LINE ENABLED                 00001
INVALID                      00000
LOGICAL ACK                  OFF
SENSE STATUS Q CNT           00
FORMAT VAR CNT               00
FORMAT VAR STRING
USER DATA NAME                 $TEST1.EUR1OBJ
USER DATA

S T A T U S:
------------

CURRENT STATUS               STARTED
CRTP TERM STAT.              STATUS SEL
LINE PROTOCOL                X25
LOGICAL STATE                STARTED
QUEUE COUNT                  00000
QUEUE STATE                  NORMAL
RETRY COUNT                  00
WARMBACKUP LOGIC             WARMSTART
STATE DETAIL                 STATE DETAIL

S T A T I S T I C S:
--------------------

ERRS                         0000000000
INPUT COUNT                  0000000000
INPUT FAILURES               0000000000
LINE PROTOCOL                00019
MAX QUEUE COUNT              00000
MSGS DROPPED TTL             0000000000
MSG DROP RATE                0000000000
OUTPUT COUNT                 0000000000
OUTPUT FAILURES              0000000000
```

5.2. Getting Actual Status of an XPNET Component

Reflex ONE24 will get the status of all XPNET components at start-up or warm-boot if configured to do so (see section 3.3.14 regarding 'Initial Status'). The XPNET graphical icons are then driven by EMS events configured to be reported to the Status Monitor screen.

This explains why Reflex is so CPU friendly as it is not constantly probing XPNET every 'N' seconds to find out that the majority of the system is fine. This will lead to an unnecessary load on the XPNET layer of BASE24 especially in the case of thousands of XPNET components. Instead, Reflex listens to the rich supply of BASE24 exception alerts. The product subsequently reports the graphical status of XPNET entities and other issues relating to other areas of the BASE24 application.

It is possible for users, however, to request the actual XPNET status of components at any time using Reflex ONE24. This can be done at object level or at group level. In the latter case, based on the way the objects have been dragged and dropped into groups, users can request the XPNET status of all components subordinate to a group. Left click on a group or object to display the floating toolbar and click on the traffic light graphical icon to get the status from the XPNET layer rather than an EMS event.



In this case, a line and a station are in a STOPPED state indicated by the red coloration. The user has the option of then bringing up the task window to start both the station and line. In the case where all objects are down, a task could be configured to start all XPNET components for a given ATM. See the next section for task configuration.

Edit the ONE24 XPNETCNF file in your data files sub-volume to change the colour representation of each of the 7 states. This is discussed in section 2.2.

6. Using Reflex ONE24 Tasks to Control XPNET Entities

6.1. Importing ONE24 XPNET Tasks

A transit file is available that contains a number of configured tasks for each type of XPNET component. Examples of the tasks contained in this file are START, STOP and ABORT BASE24 XPNET component. This can be imported into your Reflex environment. To do this, carry out the following:

NOTE: The transit file is called TRANSTSK and is contained in your Reflex data files sub-volume. If you received a separate upgrade sub-volume for Reflex ONE24 then copy the TRANSTSK file to your Reflex data files sub-volume.

1. Logon to the Tandem with the Guardian user who owns the Reflex Pathway application.
2. Volume to the Reflex object sub-volume, e.g. RFLXOBJ.
3. Run the TASKIMEX import/export utility as follows (RFLXDAT is the location of your Reflex data files):

   > RUN TASKIMEX / IN RFLXDAT.DATACONF / IMPORT * TRANSIT TRANSTSK

4. Logon to the Reflex GUI with a user who has access to Reflex set-up facilities, e.g. a Reflex ONE24 Administrator user.
5. Click on the 'Tasks' icon on the main toolbar.
6. Click on 'Window', 'Tile Vertical' options to show both the programs window and the tasks window of Reflex.

Since this is a transit file prepared at Insider Technologies, the node, volume and sub-volume locations for the task programs should be reconciled. To do this, carry out the following still in the Reflex GUI:

1. Maximise the 'Task Master – Program Definition' window.
2. Double-click on the first XPNET program entry – ***ABRTLIN***.
3. Click on the 'Definition' tab (second tab in this window). See bitmap below.
4. Change the process name if required and all Tandem node prefixes.
5. Change all entries in the Parameters aspect of the window to the on-site preferences. The 'INFILE' should stay as 'RFBMTASK' but the node, volume and sub-volume should be reconciled to your Reflex object sub-volume.

NOTE: ONE24 'RFBMTASK' is a macro that interacts with NCPCOM (Network Control Point communication) and is contained in your Reflex object files sub-volume. If you received a separate upgrade sub-volume for Reflex ONE24 then copy the RFBMTASK file to your Reflex object sub-volume.

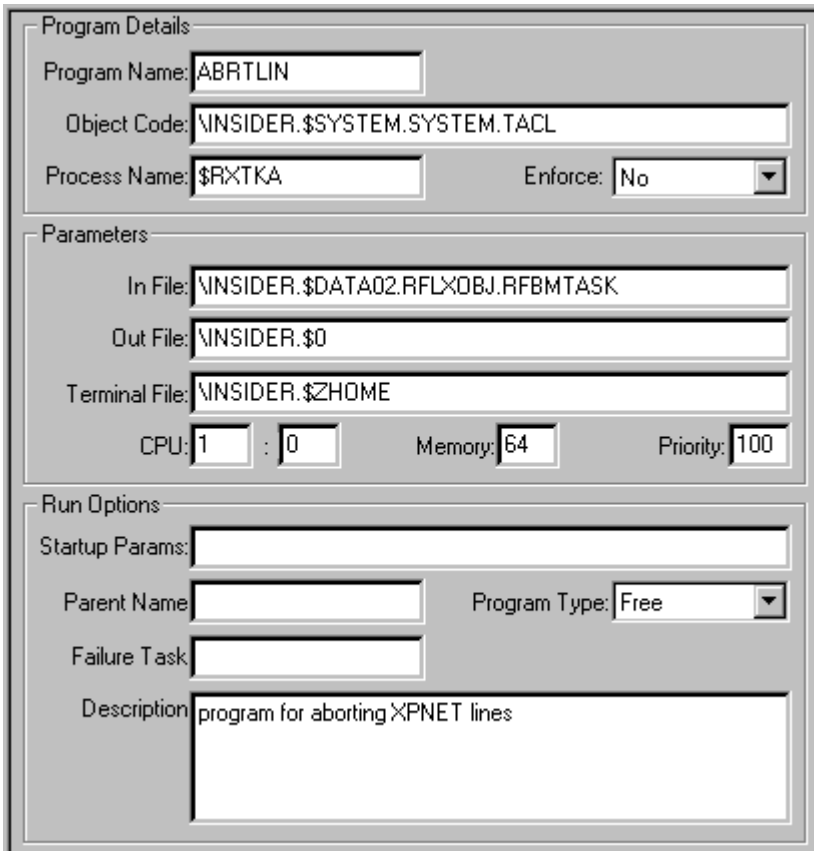
6. All other tabs should remain as delivered for the program window. Click on the tick graphic on the program window toolbar to amend the record.
7. Repeat for all other XPNET programs. Go back to the list tab for programs and look for those with XPNET with the description.

All the tasks are owned by SUPER.SUPER and will be executed by the owner of the Reflex Pathway application. If you require other users of Reflex to run these tasks or you need to run the tasks as a different user in your BASE24 environment other than the Reflex Pathway user, e.g. 'REFLEX.OWNER' then carry out the following tasks:

1. Click on the Tasks module and maximise the Tasks Master window (as opposed to the 'Task Master – Program Definition' window.
2. Double-click on the first XPNET Task – TABRTLIN.

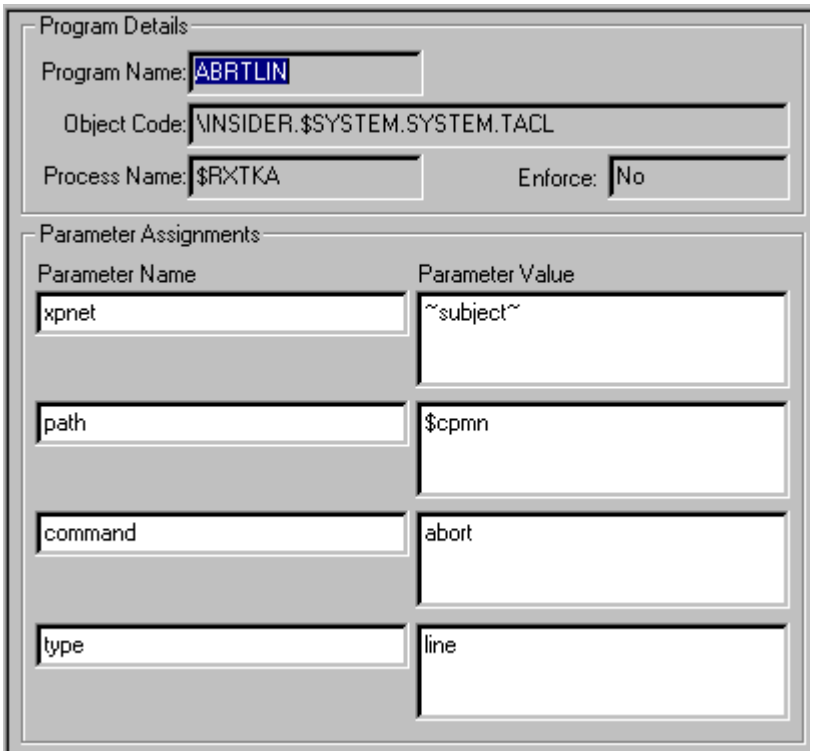
3. Click on the second tab ‘Task Configuration’ to display the program run as part of this task. Enter the ‘Execution User’ details and password. This will be the owner of the process when it runs up. Ensure the user has the authority to access NCPCOM and invoke the appropriate commands that make up this task.
4. Click on the tick icon to amend the details.
5. Click on the, ‘Task Owners’ tab (third tab from left) on this window.
6. Type the task name in the ‘Task Name’ field if not already there and give it to any other users of Reflex who may need to run the task.
7. Click on the ‘**+**’ icon to add this task owner record.
8. All other tabs should remain as delivered for the Task window.
9. Repeat for all other ONE24 XPNET tasks. Go back to the list tab for tasks and look for those with XPNET for the description.

The bitmap below shows the program details as delivered in the TRANSIT file TRANSTSK. These should be changed as described in the instructions above.



6.2. The RFBMTASK Macro for Initiating XPNET Tasks

Each ONE24 program (as described in the last section) has a parameters tab entry which contains four parameters as shown below.



The '~subject~' parameter will be substituted with either the subject token of the incoming XPNET component EMS event or the XPNET graphic icon description of the Status Monitor screen. This will depend on whether task is started automatically or manually respectively.

The remaining 3 parameters relate to:

- The live / production BASE24 Pathway
- The command to be performed
- The XPNET component type

These parameters at run time, will be fed into the ONE24 RFBMTASK macro below. This TACL macro interacts with NCPCOM using the parameters supplied.

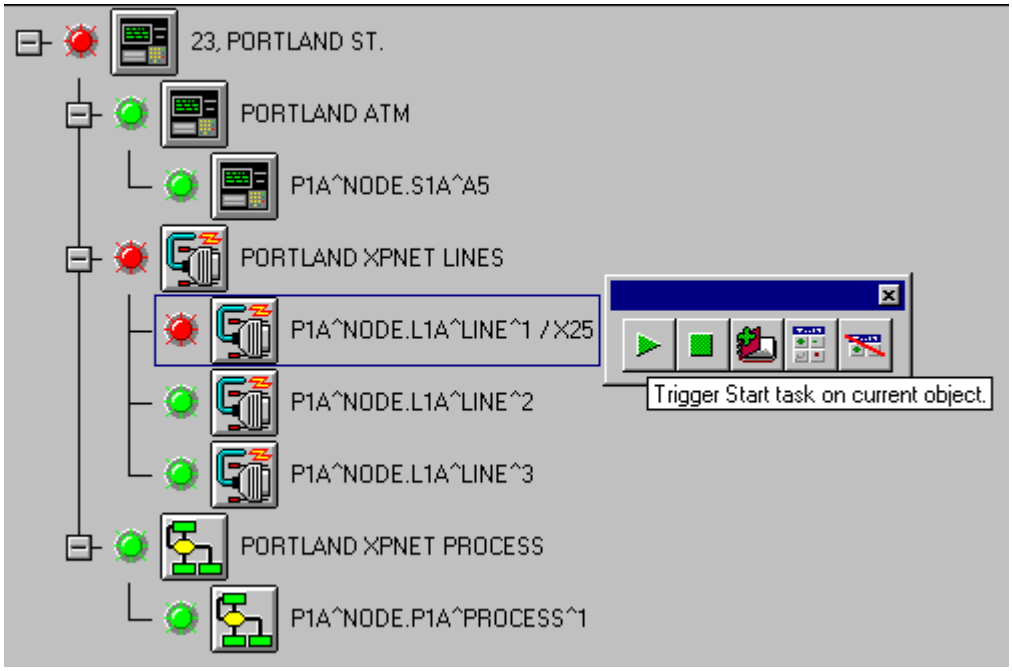
NOTE: In the case where security is switched on in the XPNET layer (see section 2.3), the ONE24 macro can be amended to supply an appropriate user to allow the task command to occur (logon command as illustrated).

```
?TACL ROUTINE

#FRAME
#PUSH #INFORMAT
#SET #INFORMAT TACL

#push object
#push path
#push command
#push type
#push :var

#set object [#param xpnet]
#set path [#param path]
#set command [#param command]
#set type [#param type]

#append :var logon reflex/invoke, insider
#append :var path [path]
#append :var [command] [type] [object]

ncpcom /inv :var /

#output Parameter is : [object]

#UNFRAME
```

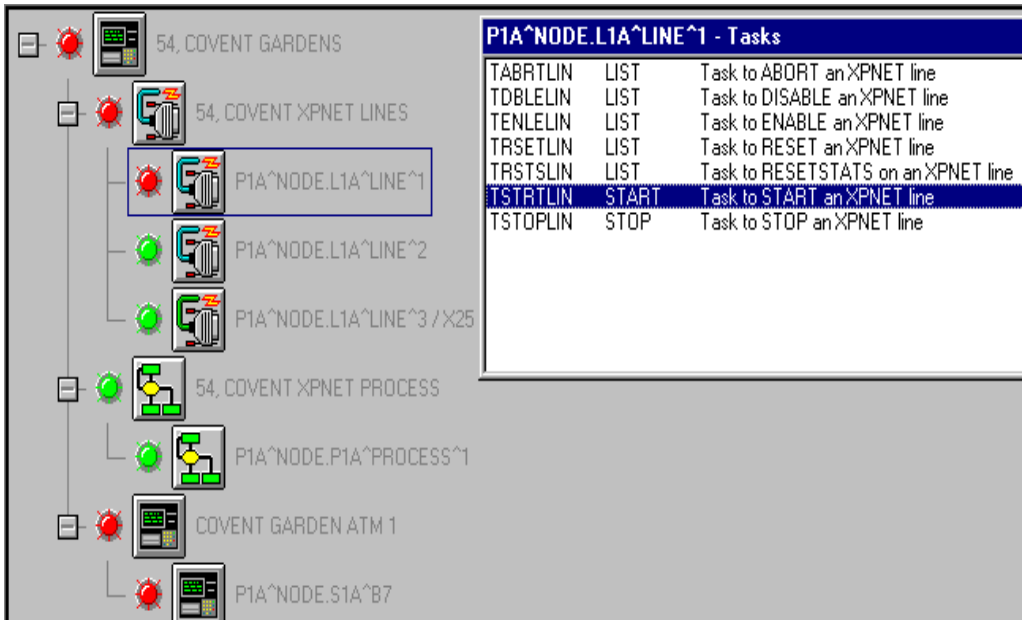
A copy of this macro can be used if requiring wild-carded commands such that the ~subject~ could be substituted in the parameters tab with a wild-carded entry. Another approach would be to change a copy of the above macro. With this approach, the NCPCOM command ‘UNDER’ could be used, e.g. ‘start station *, under node p1a^node’, ‘start station *, under line ~subject~’ and so forth.

6.3. Starting ONE24 XPNET Tasks in Reflex Status Monitor

The Status Monitor screens of Reflex avoid the need for users to logon to the Tandem platform, get a TACL session and execute XPNET component commands via NCPCOM; this can all be achieved using the TASK functionality of Reflex ONE24. An example of this can be seen in the following bitmap.



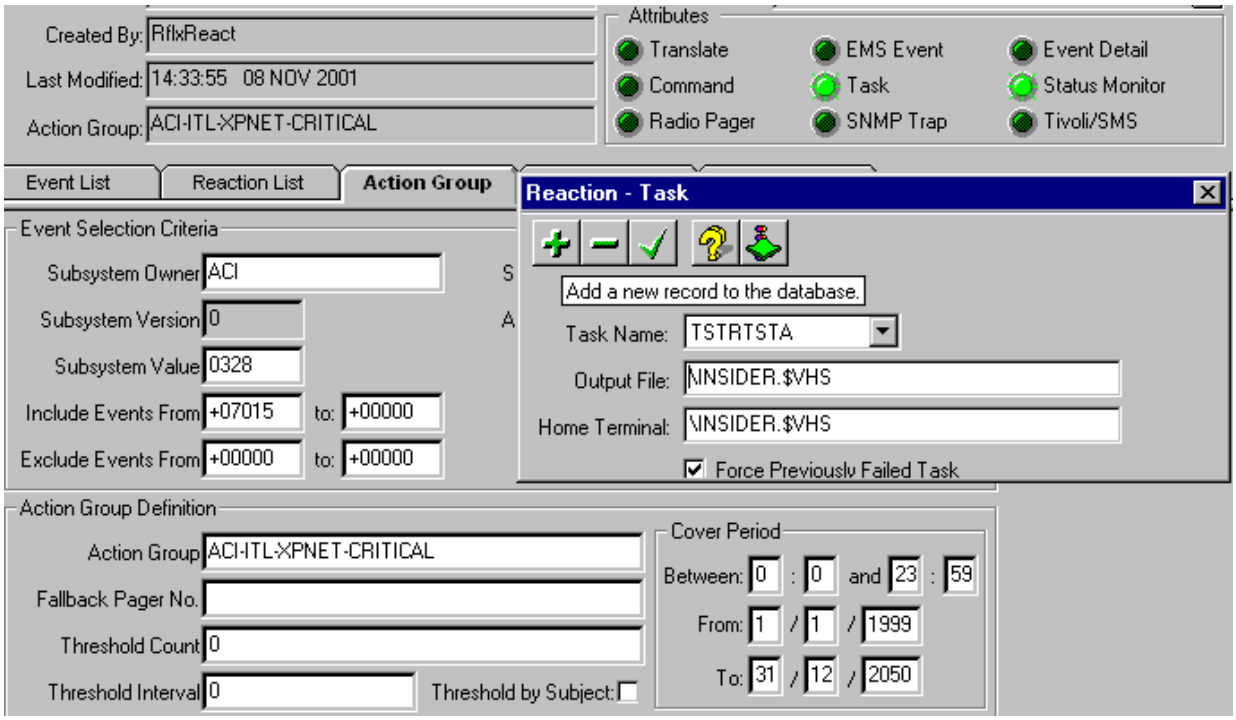
Clicking the arrow icon will initiate the task associated with a line start in the Tasks module discussed in the previous section. The stop button will initiate the stop task associated with a line. Click on the book graphical icon to display all tasks currently set-up, which relate to XPNET lines. This is shown below:



The Reflex administrator should have given all Guardian users who require to initiate ONE24 XPNET tasks, the appropriate authority via the Reflex GUI to invoke the task. This can be achieved within the task module of Reflex under the Task Owners tab of the task window.

6.4. Automatic Starting of ONE24 XPNET Tasks

Rather than using the Status Monitor screen discussed in the last section, to invoke tasks, the Reaction module of Reflex can be used to kick a task automatically on seeing a critical event. This may be especially useful when a station is stopped or aborted or a line goes down and needs to be restarted immediately. Below is a bitmap of the Reaction module and the task drop down window for associating a task with an event. A remedy for the problem is invoked without the need for operator intervention.

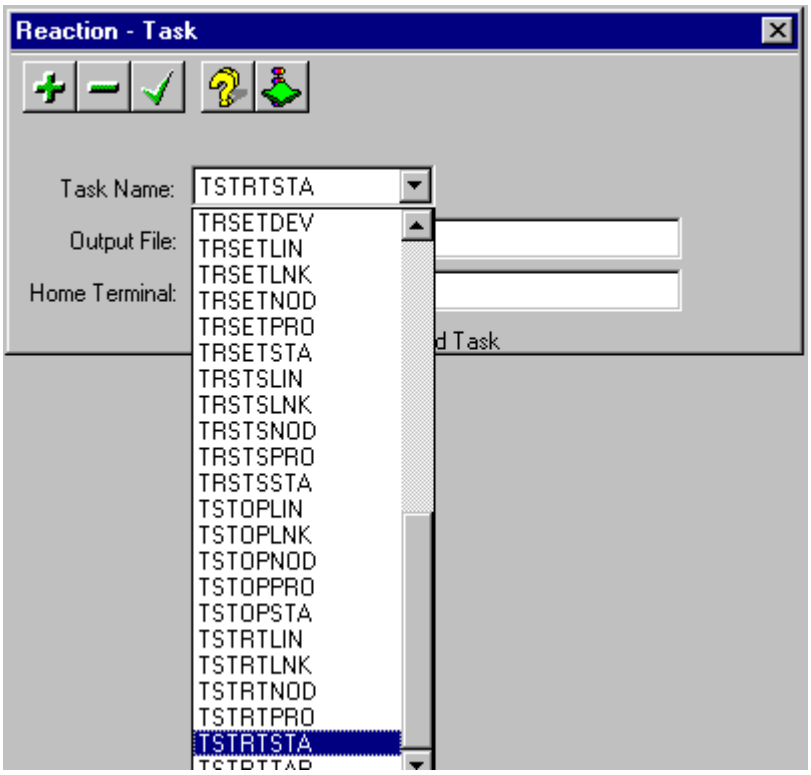


In the above example, a start station task is being associated with event ACI.328.7015 (this is only an example). On receiving this event from the XPNET layer of BASE24, Reflex will automatically start the station contained in the incoming EMS alert.

In order to add a task to an event or event range, carry out graphical monitoring:

1. Logon to the Reflex GUI with a user who has access to Reflex set-up facilities, e.g. a Reflex ONE24 Administrator user.
2. Click on the Reaction module on the main toolbar of Reflex.
3. Maximise the resulting window.
4. Click on the 'Action Group' tab of the maximised window.
5. Enter the SSID of the event(s) you wish to start a task for (NOTE: you can pick the BASE24 event from the previous tab also).
6. Enter the event or event range of the EMS event(s) you wish to monitor if not already listed.
7. Enter an Action Group. This is a free text field used to group a reaction(s) that you wish to occur when the given EMS event(s) is passed to Reflex. This Action Group can be used again if just wishing to carry out the same reaction, e.g. start a given XPNET station. Action group 'ACI-ITL-XPNET-START' could be used as the action group to start an XPNET station, line or process. This action group is subsequently associated with EMS events relating to XPNET stations in a down state.
8. Enter a desired cover period.

9. Click on the Task green attribute button that is the middle right reaction. The following window will be shown:



10. Use the drop down list to associate the task you wish to initiate on receiving the event or event range.
11. Enter your preferred output file and home terminal (see previous bitmap for appropriate syntax).
12. Click on the add button '**+**' on the window to add the record. If you are re-using the same action group in a BASE24 event range, then the details may already exist in which case just click the tick (amend) icon to amend the record. This sets a flag on the action group screen to denote that a reaction has been set-up for the given event or event range.
13. Click on the OK confirmation box and exit the above window.
14. Click on add '**+**' icon on the Reaction window (or amend - tick) to record the addition or modification.

Carry out these steps for other event or event ranges that you wish to initiate tasks for. When you have completed the new task additions, carry out the following steps:

NOTE: If just changing a task or field entry rather than an event range value, then just carry out step [10] below. This is because the filter already contains the event you are interested in from a previous generation and compilation of the filter.

1. Click on the '**!**' icon on the reaction toolbar. The following floating window will be displayed.

2. Click on 'Generate Filter Source' button and wait for confirmation of completion. This generates a filter file to pass your newly added BASE24 events to the Reflex event monitor.
3. Click OK on the confirmation message. Click on the 'Compile Filter Program' and then click OK to the 'compilation started' confirmation message.
4. Logon to the Tandem as the Guardian user who owns the Reflex Pathway application and check peruse for successful compilation listings. Alternatively, bring up CONSOLE and check for successful compilation EMS messages generated by Reflex.
5. Click on the Warm-boot Reaction Servers' to allow the event monitor to pick up the new filter for your newly added event ranges.

See the Reflex 80:20 User Manual for more details about using Reaction module.