Technical Note

NV9000-SE Utilities Revision Migration

Changes in NV9000-SE Utilities (SE for short) and corresponding changes in NV9000 controller internal software from version 4.x to 5.x to 6.x introduced some incompatibilities between versions. Customers wishing to upgrade must perform a certain amount of "manual" database conversion. These changes affect tielines and virtual crosspoints primarily.

Introduction

Revision Highlights

- At revision 4.6.12, the NV9000 software changed the way virtual crosspoints are handled. When you upgrade from early versions to version 4.6.12 or later, the NV9000 system controller loses its virtual crosspoint connections. Customers must reconnect their virtual crosspoints. Once the virtual crosspoints are rebuilt, they are retained normally. See Virtual Crosspoints on page 19.

  (Configurations of virtual crosspoints, however, do not change from revision to revision.)
- At revision 5.0.0, SE introduced multi-hop tielines. The way SE handles tielines is entirely new. During conversion, customers must recreate tieline data and create a new 'Virtual Level Signal Binding' table.

  The "preferred destinations" of SE 4.x are not supported in SE 5.x and SE 6.x.
- At revision 5.1.0, SE introduced suffix pages for NV9640, NV9641 control panels. The NV9642 has suffix pages also, but it was not introduced until revision 5.2.0.
- At revision 5.2.0, SE introduced the 'Dynamic Update' button in the 'System Management' page.
- At revision 5.2.0, SE also introduced the NV9642 and global navigation for NV9640, NV9641, and NV9642.
- At revision 6.0.0, SE changed "show names" to "name sets" and changed the way it works somewhat.

  Kaleido multi-viewers can now use name sets (at 6.0.0). This is an enhancement to the NVISION Ethernet protocol (NVEP), so theoretically other products that use NVEP could also use name sets.
- At revision 6.0.0, SE added *tieline groups*.

Misc. Notes

At revision 5.0.0, SE lost the need for a tieline server as an external interface. (Customers still need the tieline license.)

Caution: If you launch SE 5.x, it will open whatever configuration you had worked on most recently. If that configuration was a 4.x configuration with tielines, SE 5.x will — without any warning — import the old configuration, eliminate all its tieline data and recreate the configuration as a 5.x configuration. *Your 4.x configuration will be lost.* Therefore, before launching SE 5.x, make at least one backup copy of your 4.x configuration.

SE 6.x will also open and convert a 4.x configuration, but it *does* give you a warning and the chance to cancel the operation before you destroy any data.

You may contact Kieran Lyons at Miranda GVD to perform tieline conversion for you. Send him your configuration as a .zip or .xml file. He will send you a reconfigured .zip or .xml file back.

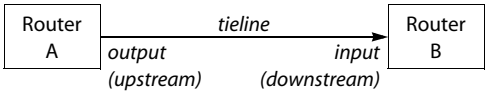
Tielines

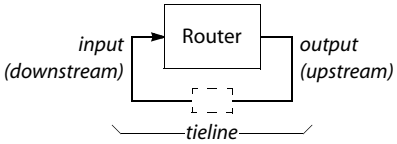
Migrating from SE 4.x to SE 5.x (or 6.x) requires a complete reconstruction of tieline data. It is imperative to make backup copies of your 4.x database.

Migrating from SE 5.x to SE 6.x requires is straightforward, automatic, and requires manual conversion effort. Note SE 6.x supports *tieline groups* which were not available in SE 5.x.

Tieline Principles

Simply stated, a tieline connects two routers. More precisely, a tieline connects an output of an "upstream" router partition (*physical level)* to an input of a "downstream" router partition. The upstream and downstream router partitions can be the same.

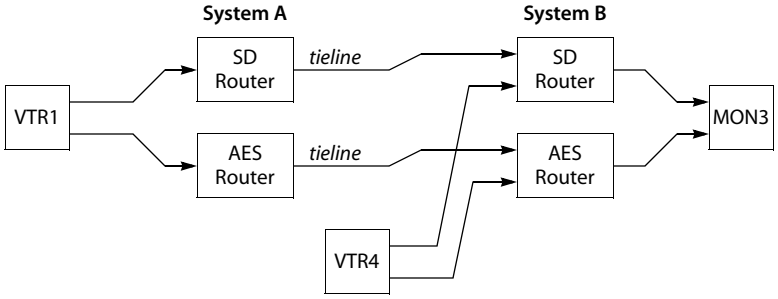




The following examples show some uses of tielines.

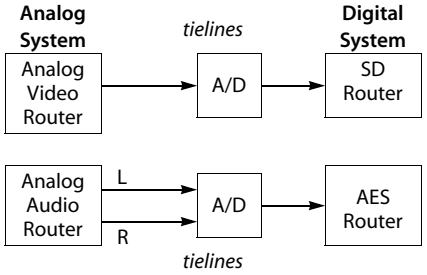
- Case A: connecting two routers.

  In this case, tielines connect two different router systems. VTR 1 is connected (only) to routers in System A. MON 3 and VTR 4 are connnected to System B. With tielines, you can take VTR 1 to MON 3. Tielines are not needed to take VTR 4 to MON 3.



- Case B: format conversions.

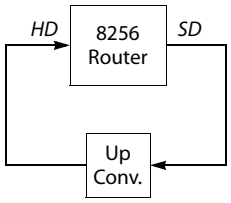
  In this case, tielines use intermediate devices to connect router matrices of different types. Here, analog signals are converted to digital signals along tielines:



  Again, it is the Tieline Manager that makes these connections useful. With tielines, it is possible to take an analog source to a digital output (in one take on a single control panel).

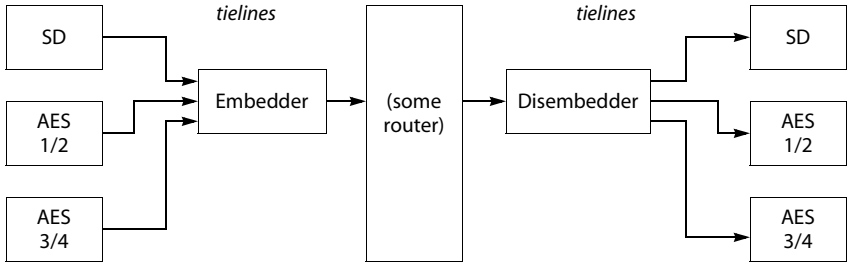
- Case C: up- or down-conversions.

  In this example, a tieline connects ports in the same matrix:



- Case D: embedding or disembedding.

  In this example, a tieline connects multiple ports at one end and a single port at the other end:



Discussion

These are the main reasons for tielines:

- Switch a source on one router to a destination on another router.
- Connect routers in separate locations.
- Expand smaller routers.
- (Commonly) to perform signal conversions.
- (Commonly) to perform audio embedding or disembedding.
- Create signal delays (for audio).

These are the specific benefits of tielines:

- Only a single take is required.
- Tielines for a take are obtained from a pool of tielines that you define. Ports are not committed to particular tielines. Takes using tielines (as needed) are transparent to the operator.
- You can define ‘strict’ or ‘flexible’ tieline usage.

Tielines in SE 4.x.x

An NV9000 tieline server is needed at revision 4.x.x. It is no longer needed at SE 5.x.x (and later). The tieline server as added as an external interface. (A license is always required for tieline service on the NV9000—at any revision of SE.)

Tielines should belong to tieline *classes*. The tielines themselves are defined as *device connections* and as such belong to level sets that have inter-level-set mappings. The tieline classes distinguish different kinds of tielines. For instance, if you are using tielines to represent up- and down conversion paths, you would have two classes, one for up conversion and one for down conversion.

There is a configuration page for defining tielines. When you are using this page, a yellow box represents the upstream level set and a blue box represents the downstream level set. You draw arrows to represent the connections on the virtual levels. You can specify a tieline connection method (none, strict, flexible) for each

virtual level. At the right of this page, you specify (optionally) tieline classes and the actual physical tielines themselves — as device to device connections. (The devices will have already been defined, at an earlier stage, with tielines in mind.)

Tieline Types

Tielines use inter-level-set mappings. Inter-level-set mappings can be:

- 'Strict', where the specification is rigidly enforced. That is, the only virtual levels the tieline may use are the exact levels defined in it.
- 'Flexible', where the software allows either the defined levels or any of their physical equivalents to use tielines having that specification. (Virtual levels are physically equivalent if the inputs and outputs concerned are on the same physical routing matrix.)
- Neither of these. The inter-level-set mapping does not apply to tielines.

Summary for SE 4.x.x

- Tielines connect devices.
- Tielines belong to classes. The tielines in a class are pooled or shared when a tieline of that class is needed.
- Tieline classes have "preferred destinations."

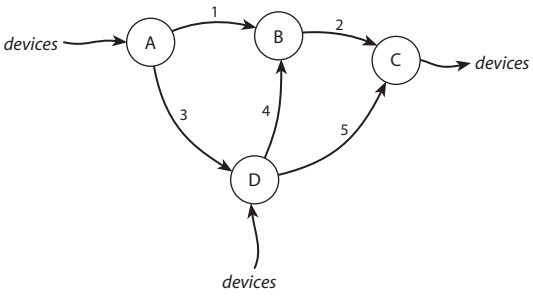
The concepts of inter-level-set mapping, of strict and flexible usage, and of "preferred destinations" are gone in SE 5.x and 6.x.

Tielines in SE 5.x.x

At revision 5.x.x, SE no longer has inter-level-set mappings and does not have tieline classes.

Tielines are no longer *device* connections, but *port* connections. (A port is a router input or output, often having a BNC connector.) For tielines, there are no devices: no sources and no destinations and, therefore, no "preferred" destinations.

SE 5.x.x allows multi-hop tielines:



What "multi-hop" implies is that you can take a source from a router at point A to a destination at point C (in a single take). There are several paths from A to C: the NV9000 control software chooses the best path (usually the shortest and of the least cost).

No special configuration need be performed to create multi-hop tielines. It is sufficient to configure just the individual port connections.

When you write the configuration to your NV9000 system controller, SE analyzes all the tieline paths that can possibly be made. (The process takes a few seconds prior to the write.) A take requiring tieline(s) will follow the shortest or lowest cost path from its source device to its destination device.

The new tieline methods require "virtual level signal bindings" which are usually created automatically. However, these signal bindings do not exist in SE 4.x and must be created manually at the time you convert a 4.x configuration to a 5.x configuration.

The methods of creating tielines in SE 5.x is different from the methods of SE 4.x.

SE 5.x has added a ‘Tieline Status’ page (under System Management). This page shows graphically tieline topology and counts for each tieline in a running system. The page also has a table of current tieline data.

SE 5.x uses the term “Tieline2” in the tables. “Tieline2” means “Tieline”—the reason for the naming has no bearing on tielines.

SE 5.x has tieline tables that exist for tieline status: ‘Tieline2 Path’, ‘Tieline2 Path Details’, and ‘Tieline2 Reservations’. (A reservation occurs when a tieline is selected and used.) These tables cannot be populated by the configurer and are not relevant to conversion of SE 4.x tielines to SE 5.x tielines.

All the other tieline tables (e.g., Tieline2 Signal Binding) are generated automatically when you create tielines.

Tielines in SE 6.x.x

The treatment of tielines in SE 6.x.x is exactly the same as it is in SE 5.x, but with the addition of (1) tieline groups and (2) multi-site tielines.

In SE 6.x.x, a *tieline group* couples tielines that must be used together.

Tieline classes in SE 4.x supported “preferred destinations” whereas in SE 5.x and 6.x tielines do not involve destinations and therefore the concept of preferred destination is gone.

With multi-site tielines, tielines defined in one NV9000 system controller can use tielines defined in another NV9000 system controller. The different NV9000s can perhaps be in different cities. The tieline definitions in each NV9000 must include definitions for the tielines in the other NV9000s. This requirement warranted the addition of the ‘Control Systems’ table where the different NV9000s can be enumerated.

Additional changes that might relate to tielines are:

- The ‘Remote’ table is gone. This table was not used in SE 5.x.
- There is now a ‘Device Physical Connections’ page under Configs. It is very useful, but does not affect tieline upgrades.
- There is now a ‘Find Router’ page under Tasks.

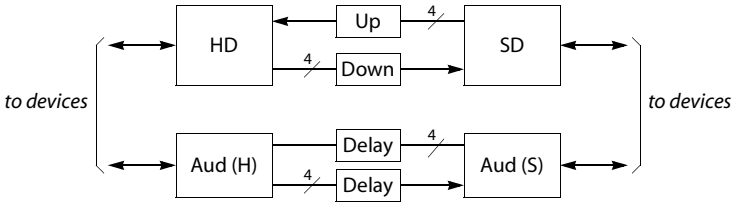
Tieline Migration

(1) Manual tieline conversion is required only for NV9000 systems going from SE 4.x to 5.x or from SE 4.x to 6.x. Conversion of tielines from SE 5.x to 6.x is automatic.

(2) Tielines in SE 4.x are neither multi-hop nor multi-site. Therefore, you need not consider those topics when converting to SE 5.x or SE 6.x.

A (Simple) Migration Example

In this example, an HD and audio router combination in one building exchanges signals with an SD and audio router combination in another building. Between the routers, up-conversion uses 4 sets of tielines and down-conversion also uses 4 sets of tielines:



There are two tieline classes: ‘HD to SD’ and ‘SD to HD’.

Each set of tielines comprises one video connection and one audio connection. A delay in the audio paty compensates for the delay introduced by the down- or up-conversion.

Migration Procedure

Follow these guidelines:

1 The first step is to make a backup copy of the 4.x configuration. It is also advisable to back up all 4.x configurations. (Exporting to a .zip file is a good way to do that.)

It is imperative that your NV9000 software versions match the revision of SE you are using. In migrating to SE 5.2.4, for example, you must also upgrade the NV9000 to revision 5.2.4 software. The same is true for SE 6.x.

2 Take screenshots of all your 4.x configuration pages or otherwise make a record of ***every*** piece of configuration data.

3 Launch SE 5.x (or 6.x). Open the configuration you want to migrate. The updated configuration will retain everything but tielines (and virtual crosspoints—another matter). Verify that your routers, level sets, categories, devices, etc. are all present and correct.

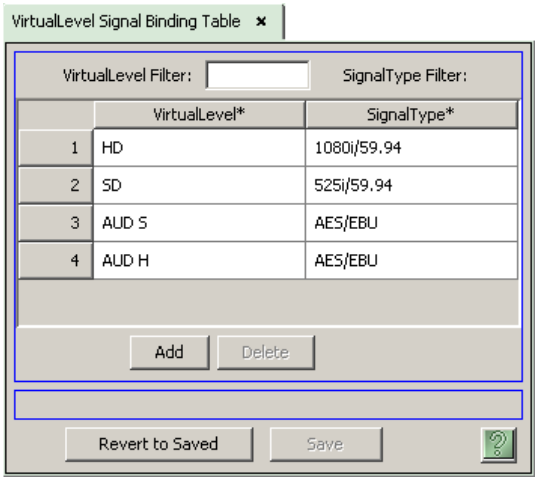
▲ Be careful: launching SE 5.x opens whatever configuration you had open last. If that was a 4.x configuration you want to preserve, it will no longer be available in SE 4.x because it will have been converted to SE 5.x. SE 6.x asks permission before it converts an old configuration.

4 Whatever devices you had created for tielines in SE 4.x, you must now delete in SE 5.x or 6.x. In the migration example, the names of tieline devices begin with “HDSD” and “SDHD.” In the ‘Devices’ table, you would select those and click the ‘Delete’ button. You could also do that in the ‘PhysConns’ table.

In SE 5.x or 6.x, tielines do not use devices, but use *ports*. Deleting the devices used for tielines frees the ports used by the devices. Now the same ports can be assigned to the tielines as you (re)create them.

5 Create virtual level signal bindings for your virtual levels. It is not difficult. Go to the ‘Virtual Level Signal Bindings’ table under ‘Views’. For each virtual level in your configuration, add a row to the table, assigning a signal type to the virtual level. The signal types are available in drop-down menus.

In the migration example, the virtual levels’ signal bindings would look like this when completed:

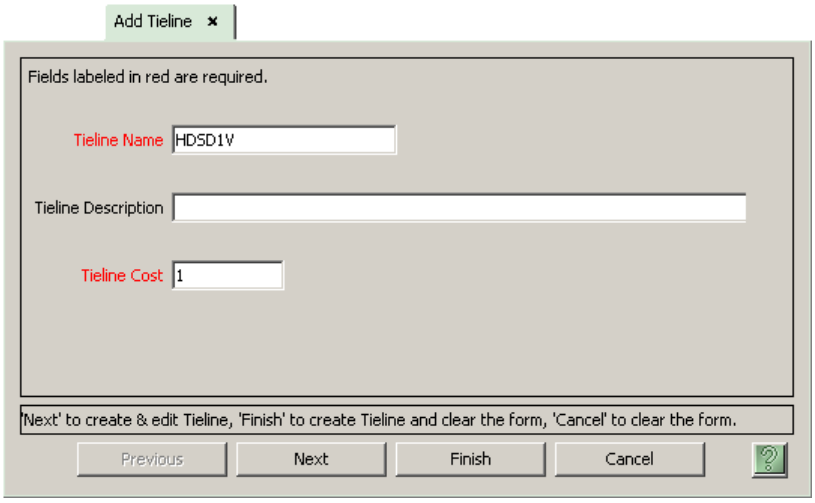


Here, we are assuming that “audio” in the example is AES. The example has just one audio level; real systems would most likely have several AES levels.

6 Once you have virtual level’s signal binding established, you can proceed to the tielines themselves. In the example, there are 4 tieline sets (8 tielines) for up-conversion and 4 tieline sets (8 tielines) for down-conversion. To recreate that in SE 5.x (or 6.x) you will create 16 tielines. The name of a tieline now corresponds to a set of ports. In the migration example, there will be 16 names.

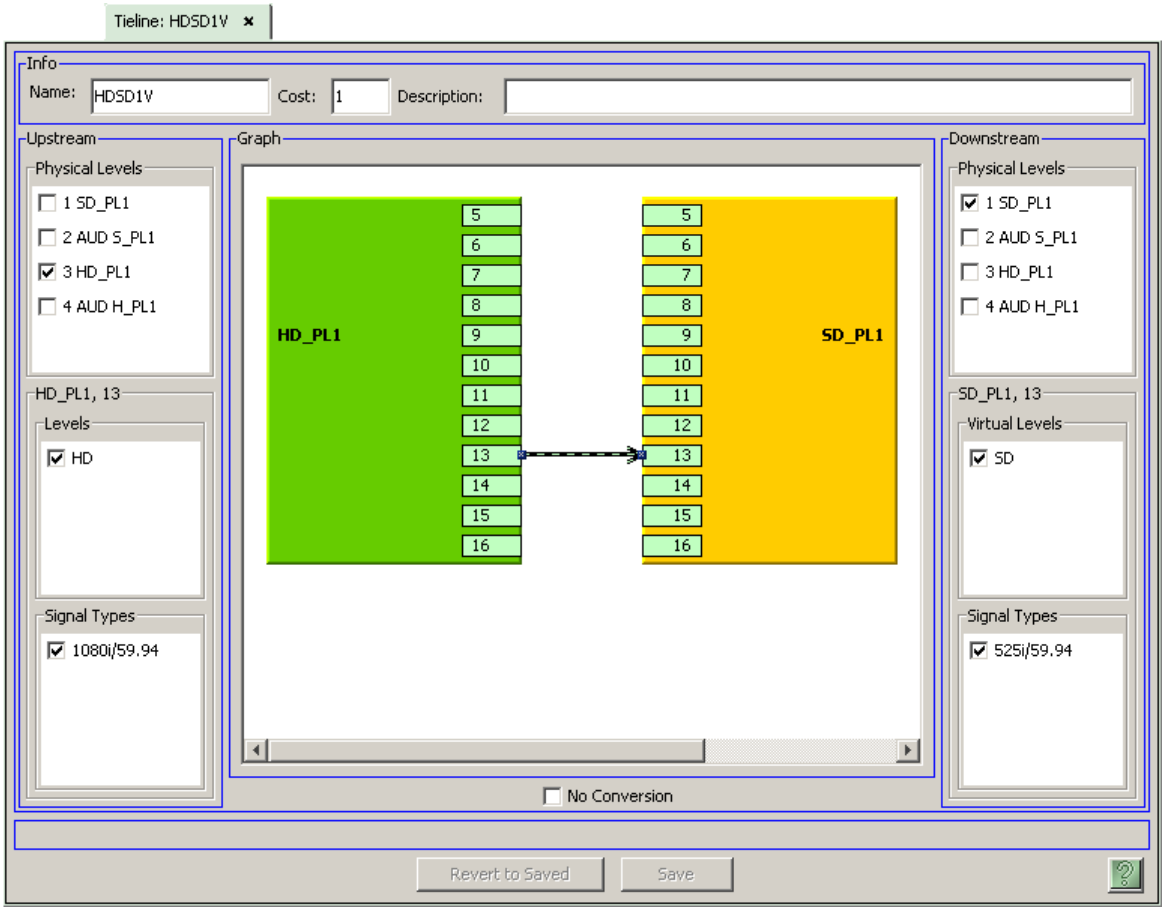
Go to the ‘Tielines’ page under’ Config’ and follow these steps:

a Click the ‘Add Tieline’ button. The ‘Add Tieline’ page appears:



Name the tieline. Then click ‘Next’.

b A tieline definition page appears:



(This is the first tieline for down-conversion in the example.)

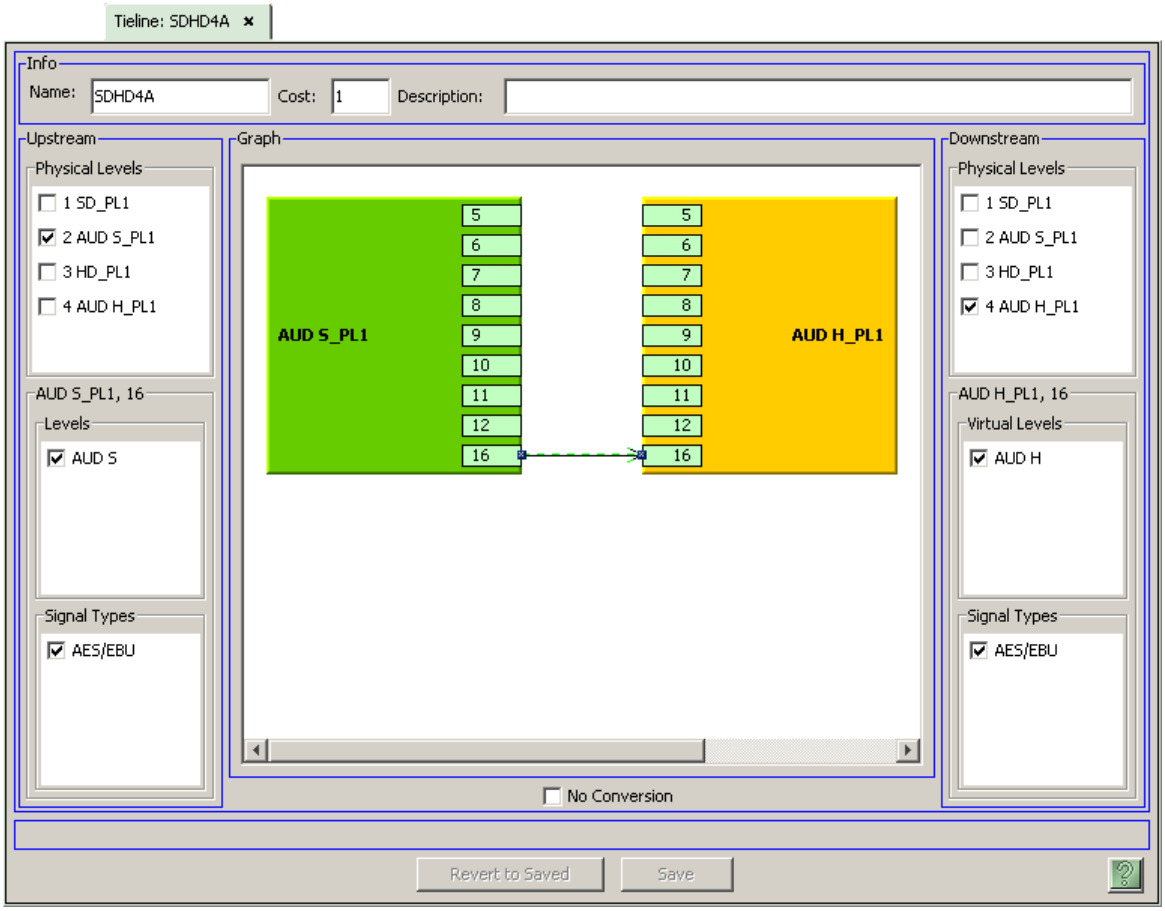
On the left, choose upstream physical level(s). On the right, choose downstream physical level(s). In the migration example, you would choose exactly one level on each side. Draw a line from an upstream (output) port to a downstream input port. Only ports that are available (not used by devices or other tielines) are displayed.

After you draw the line, choose the virtual levels on each side that apply. In the migration example, the upstream virtual level is HD and the downstream virtual level is SD. The signal types are displayed for the virtual levels.

When the tieline is defined, click ‘Save’. To cancel, click ‘Revert to Saved’.

Close the tieline definition window. Repeat steps a and b for your other tielines.

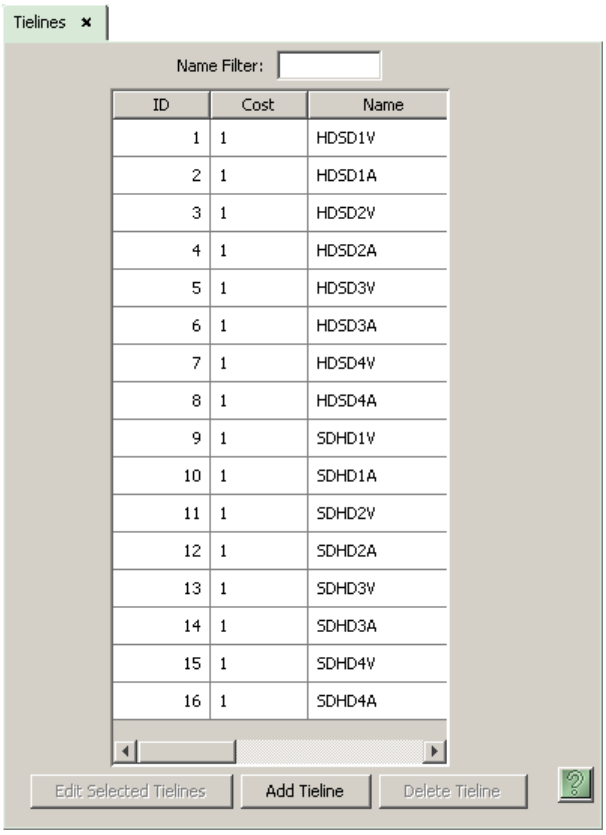
This illustration shows the definition of the last audio tieline (for up-conversion):



In the migration example, for up-conversion, the upstream level is ‘Aud S’ and the downstream level is ‘Aud H’. Note that the signal types are the same (AES/EBU) on both sides.

Because this is the last tieline configured, some of the ports have been consumed already. Absent from the green and orange graphics are ports 13, 14, and 15 on each side. (Ports 1–4 were consumed by device connections.)

7 You will have completed the conversion after you have defined all the tielines according to step 6. This is the completed tieline list for the migration example:



ID	Cost	Name
1	1	HDSD1V
2	1	HDSD1A
3	1	HDSD2V
4	1	HDSD2A
5	1	HDSD3V
6	1	HDSD3A
7	1	HDSD4V
8	1	HDSD4A
9	1	SDHD1V
10	1	SDHD1A
11	1	SDHD2V
12	1	SDHD2A
13	1	SDHD3V
14	1	SDHD3A
15	1	SDHD4V
16	1	SDHD4A

If you are using SE 6.x, you can go on to create tieline groups.

You can also assign a “cost” to each tieline now. Tieline costs were not part of SE 4.x. A tieline cost is a integer that gives a relative weight to the tieline. The NV9000 system controller will attempt to use a tieline path of the least cost.

You can use costs in a variety of ways. One use for costs would be to assign a preference to certain tielines or tieline paths. The “preferred” tielines (or paths) would have lower costs.

You could also have designated a cost at the time you created the tieline in the ‘Add Tieline’ page. You can edit the costs at any time in the ‘Tieline2’ table under ‘Views’.

8 Multi-site tielines were not part of SE 4.x so they are not a conversion issue. But they are a feature of SE 6.x.

Tieline Groups

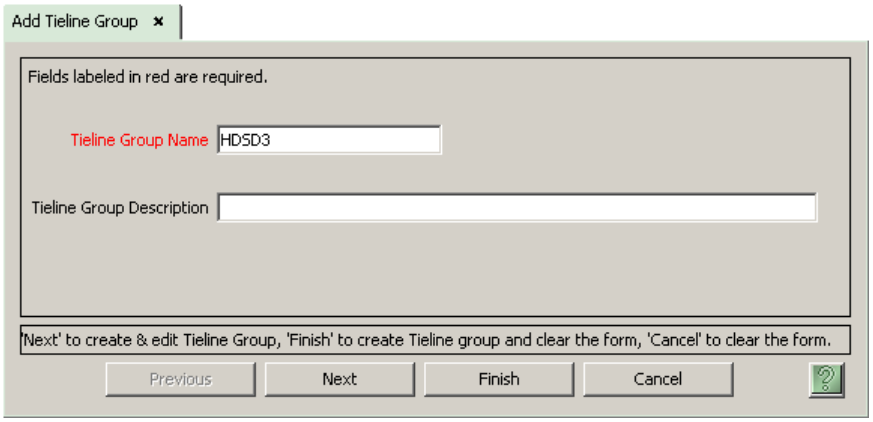
SE 6.x supports tieline groups. SE 5.x does not. And, as stated earlier, SE 4.x does not.

A tieline group bundles together a set of tielines so that if one tieline in the group is used, all of the tielines in the group are used. That is, the tielines in the group are allocated together when the need for the tieline arises.

In the migration example, each of the original tielines had a video connection and an audio connection. To bind the video and audio together, you would create 4 tieline groups for up-conversion and 4 tieline groups for down-conversion, each comprising video and audio.

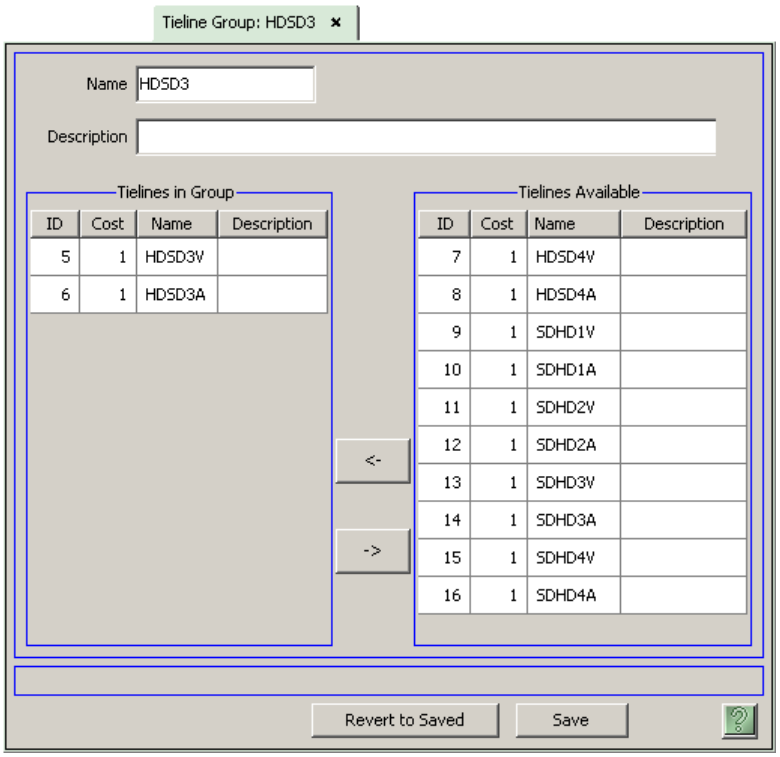
To create tieline groups, follow these steps:

1 Click ‘Tieline Groups’ under ‘Config’ and then click the ‘Add Tieline Group’ button. The ‘Add Tieline Group’ page appears:



Name the group and click ‘Next’.

2 The tieline group definition page appears:

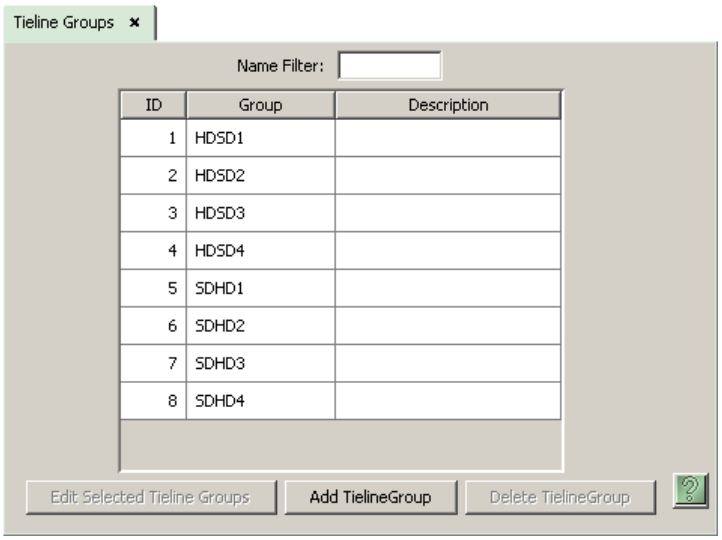


To add tielines to the group, choose some of the tielines from the right side (tielines available) and click the left arrow. The illustration shows the creation of a group for the third HD-to-SD conversion tieline.

When you add tielines to a group, they become unavailable for other groups. In this example, tieline groups HDSD1 and HDSD2 were created using tielines HDSD1V, HDSD1A, HDSD2V, and HDSD2A. Therefore, those tielines do not appear in the ‘Tielines Available’ list.

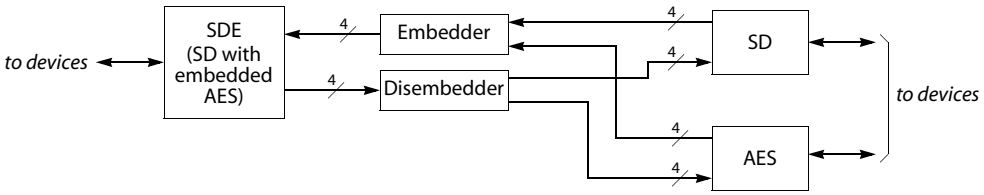
To remove tielines from the group, choose some or all of the tielines on the left (in the group) and click the right arrow. When you have finished with this group, click ‘Save’. To cancel any pending changes, click ‘Revert to Saved’.

3 This would be the result of creating 8 tieline groups, one for each of the 8 tieline sets from the original configuration in SE 4.x:



Tielines for Embedding and Disembedding

This second example presents an SD router having embedded audio that connects, through disembedders and embedders, to a separate SD router and a separate AES router. Disembedding uses uses 4 sets of tielines and embedding also uses 4 sets of tielines:

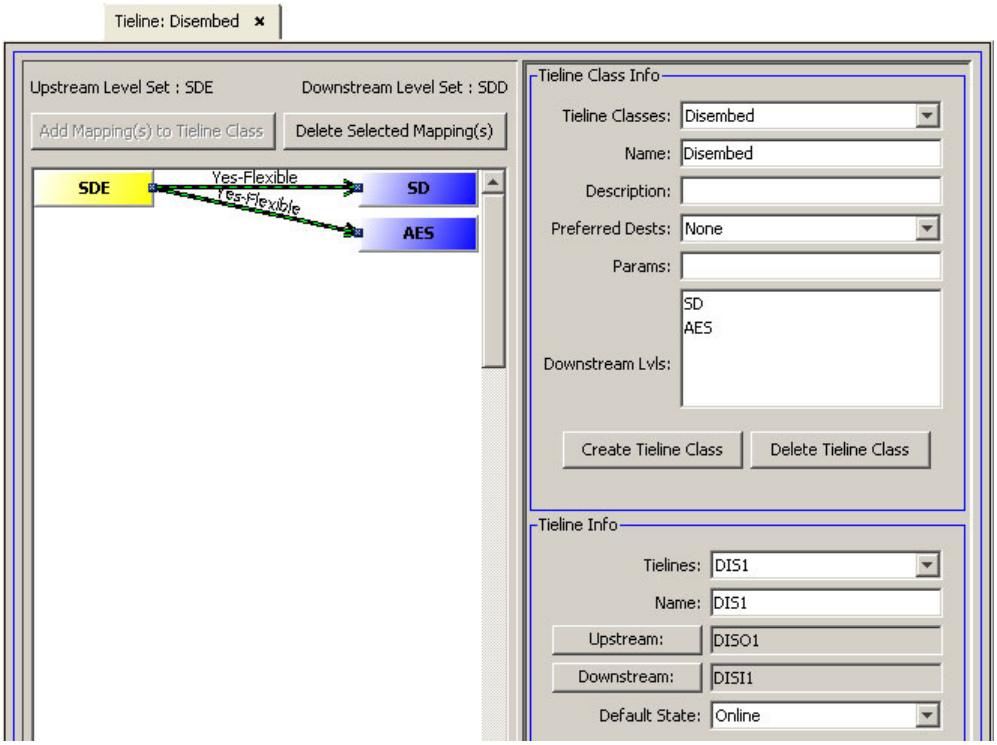


(In SE 4.x) there are two tieline classes: ‘DIS’ and ‘EMB’.

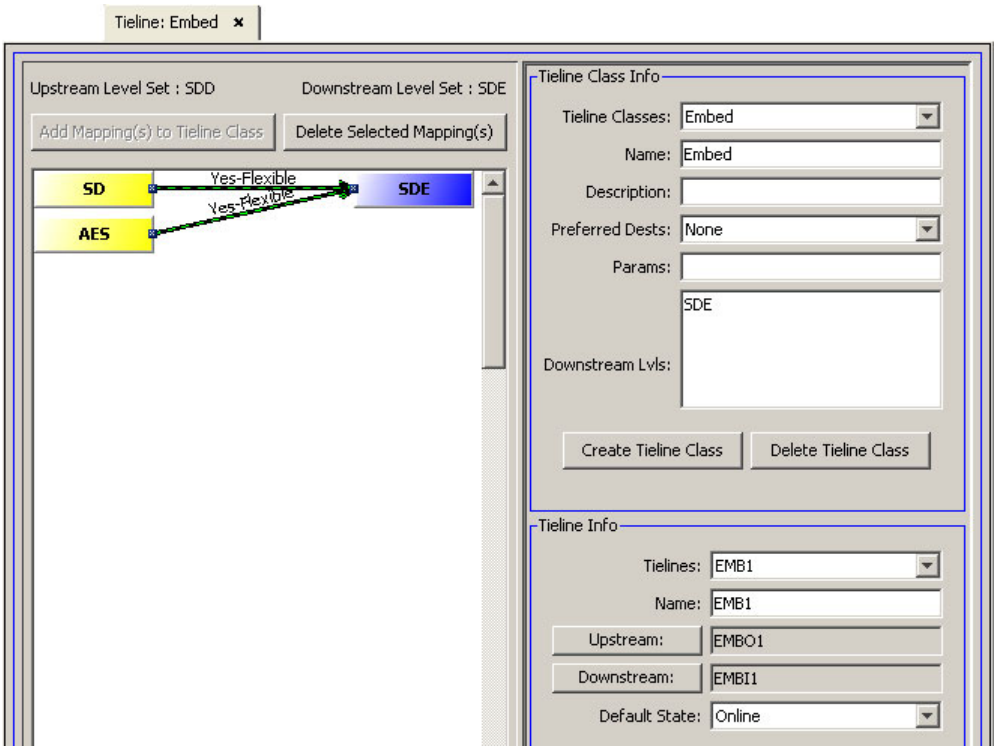
Each set of tielines has 3 connections; one for SD video with embedded audio, one for SD by itself and one for AES by itself.

To convert a 4.x configuration with disembedding tielines to SE version 5.x or 6.x, follow the same steps given earlier under Migration Procedure (page 6). When you recreate the tielines, the tieline configuration pages in SE 5.x or 6.x also show fan-out (page 14) or fan-in (page 15).

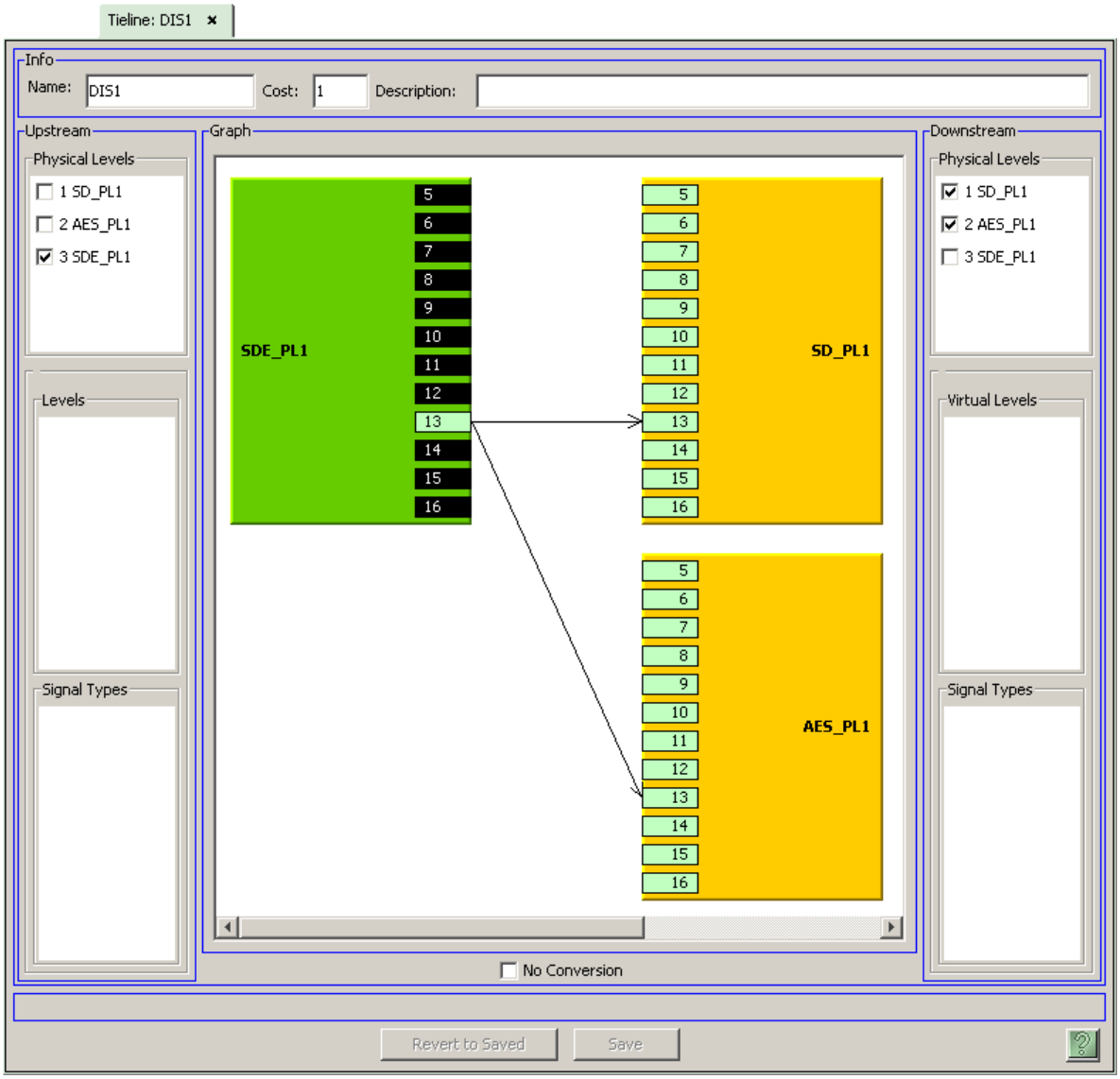
In SE 4.x, the tieline configuration page looks like this for the disembedder tielines:



The tieline configuration page looks like this for the embedder tielines:



This is the tieline page in SE 5.x for the first of the disembedding tielines, showing fan-out:

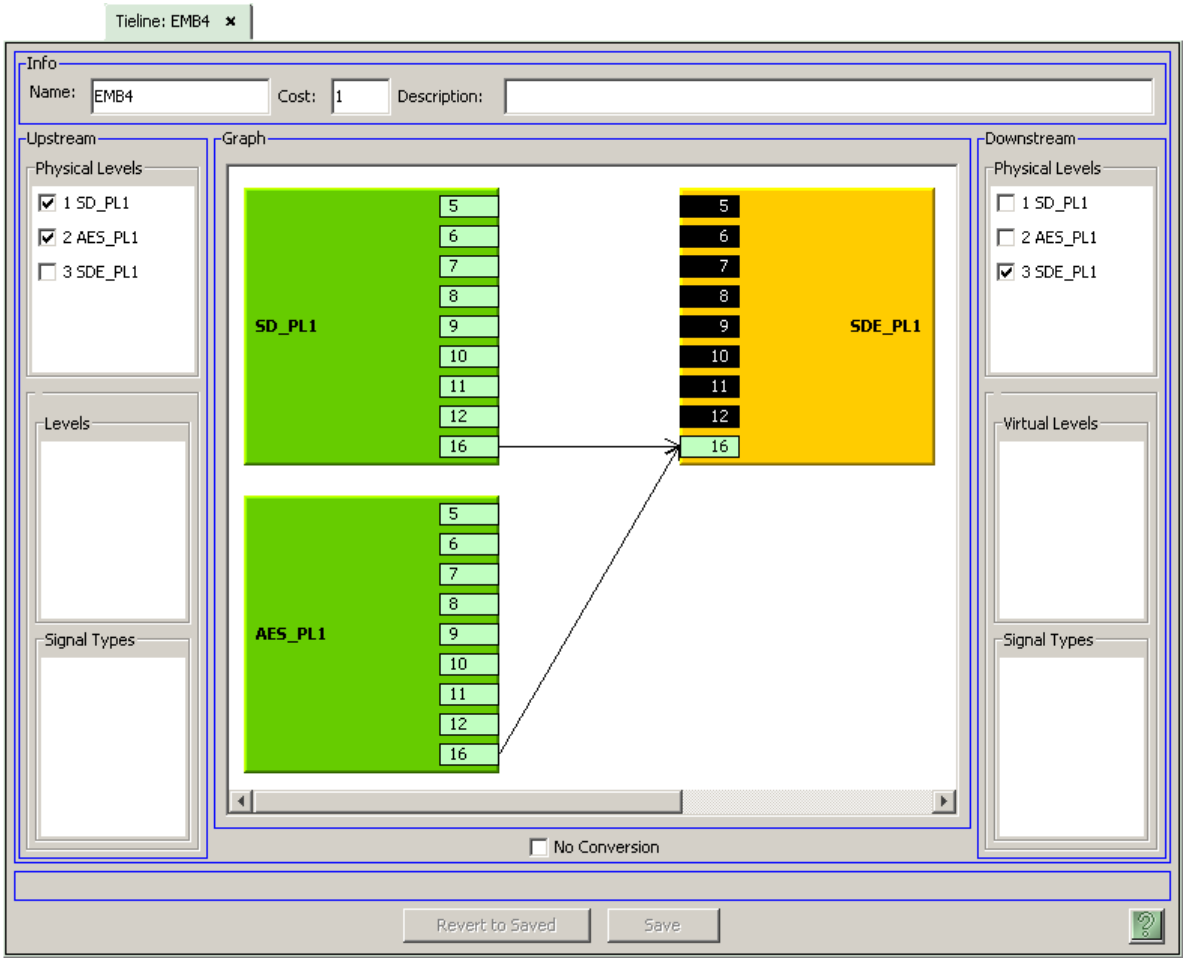


Here, the small black rectangles and light green rectangles are router ports. The black rectangles indicate that no connection is possible at that port. Connections are possible only at the ports represented by light green rectangles.

▲ Fan-out is 1:N and can never be M:N.

This example uses small routers (16×16). The first 4 ports of each router were consumed by devices (cameras, VTRs, monitors). Thus, ports 5–16 on each router are available for tielines.

This is the tieline page in SE 5.x for the last of the embedding tielines, showing fan-in:

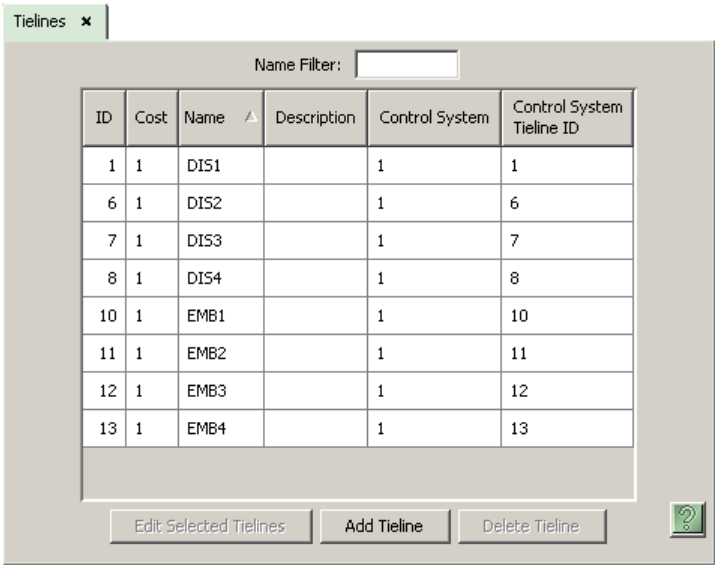


Here too, the small black rectangles and light green rectangles are router ports. The black rectangles indicate that no connection is possible at that port. Connections are possible only at the ports represented by light green rectangles.

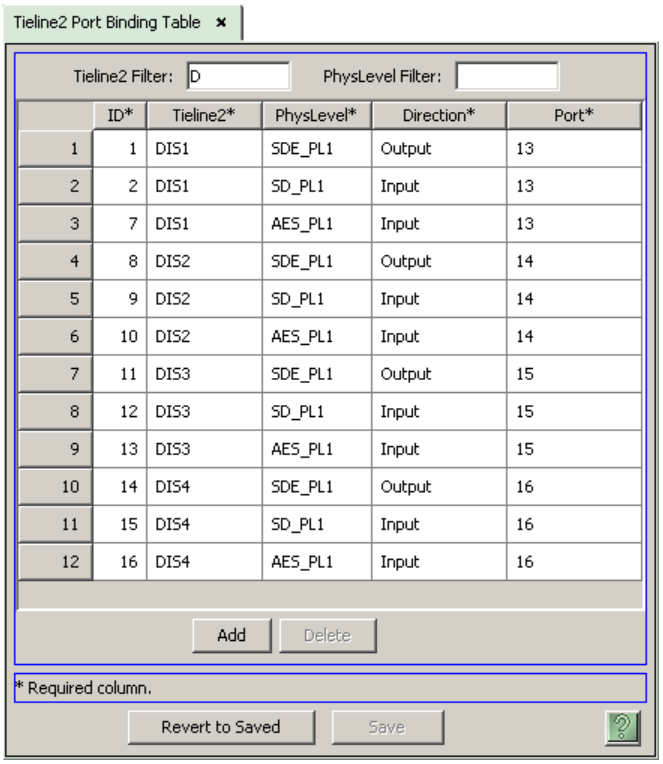
- Fan-in is N:1 and can never be N:M.

Because this is the last tieline configured, some of the ports have been consumed already. Absent from the graphic are ports 13, 14, and 15 on each side. (Ports 1–4 had already been consumed by device connections.)

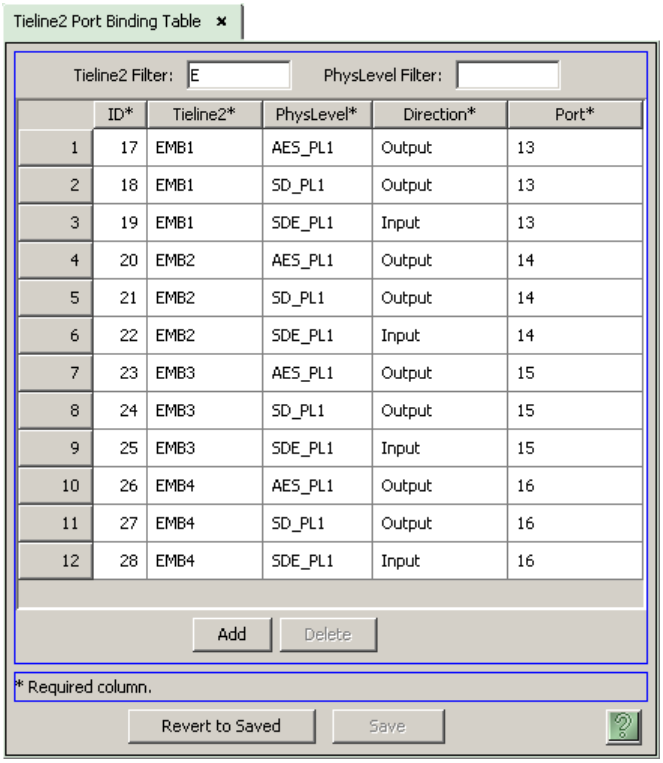
This is the completed tieline list for the embedding/disembedding example:



The tieline port binding table will have been filled automatically. This shows the 4 *disembedding* tielines:

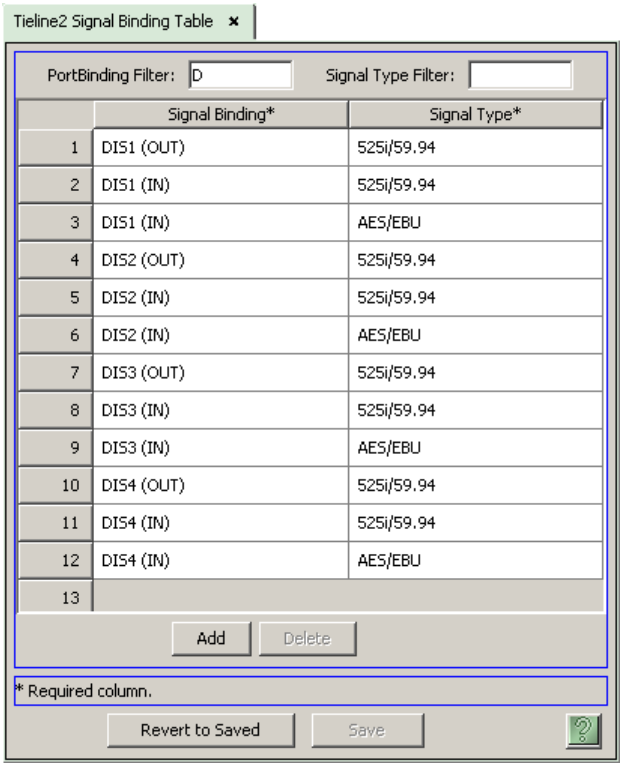


This shows the 4 embedding tielines:



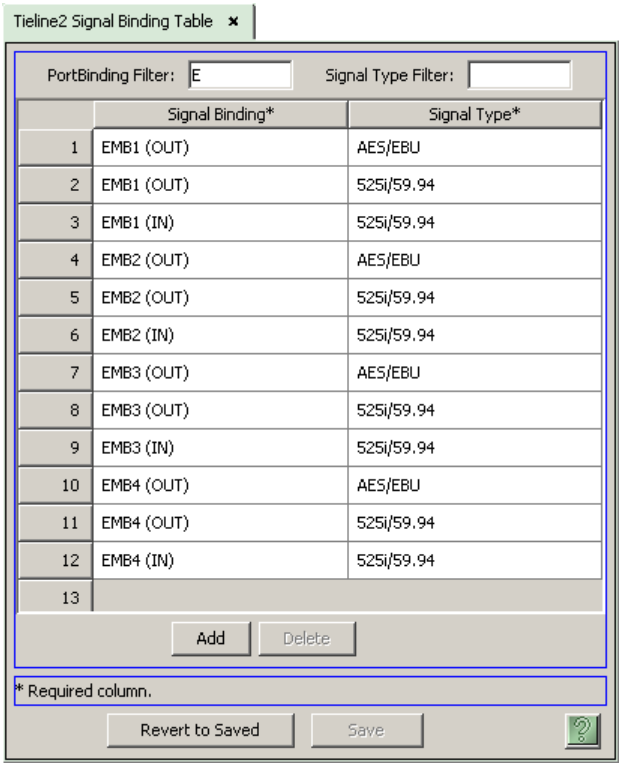
	ID*	Tieline2*	PhysLevel*	Direction*	Port*
1	17	EMB1	AES_PL1	Output	13
2	18	EMB1	SD_PL1	Output	13
3	19	EMB1	SDE_PL1	Input	13
4	20	EMB2	AES_PL1	Output	14
5	21	EMB2	SD_PL1	Output	14
6	22	EMB2	SDE_PL1	Input	14
7	23	EMB3	AES_PL1	Output	15
8	24	EMB3	SD_PL1	Output	15
9	25	EMB3	SDE_PL1	Input	15
10	26	EMB4	AES_PL1	Output	16
11	27	EMB4	SD_PL1	Output	16
12	28	EMB4	SDE_PL1	Input	16

The tieline signal binding table will have been filled automatically. This shows the 4 *disembedding* tielines:

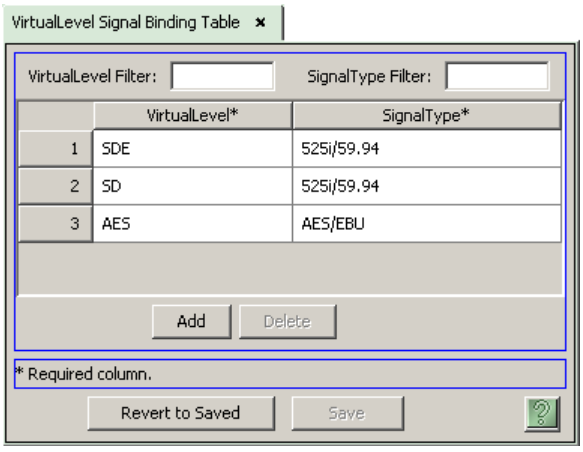


	Signal Binding*	Signal Type*
1	DIS1 (OUT)	525i/59.94
2	DIS1 (IN)	525i/59.94
3	DIS1 (IN)	AES/EBU
4	DIS2 (OUT)	525i/59.94
5	DIS2 (IN)	525i/59.94
6	DIS2 (IN)	AES/EBU
7	DIS3 (OUT)	525i/59.94
8	DIS3 (IN)	525i/59.94
9	DIS3 (IN)	AES/EBU
10	DIS4 (OUT)	525i/59.94
11	DIS4 (IN)	525i/59.94
12	DIS4 (IN)	AES/EBU
13		

This shows the 4 *embedding* tielines:



For this example, you would have created the virtual level signal binding table as shown:



Other tieline tables (such as ‘Tieline2 Paths’ and ‘Tieline2 Reservations’) are not important during configuration. These become filled if you read the configuration from a running system that has tielines. (The running system should have the same configuration as the one with which you are working if the retrieved tieline data are to make any sense.)

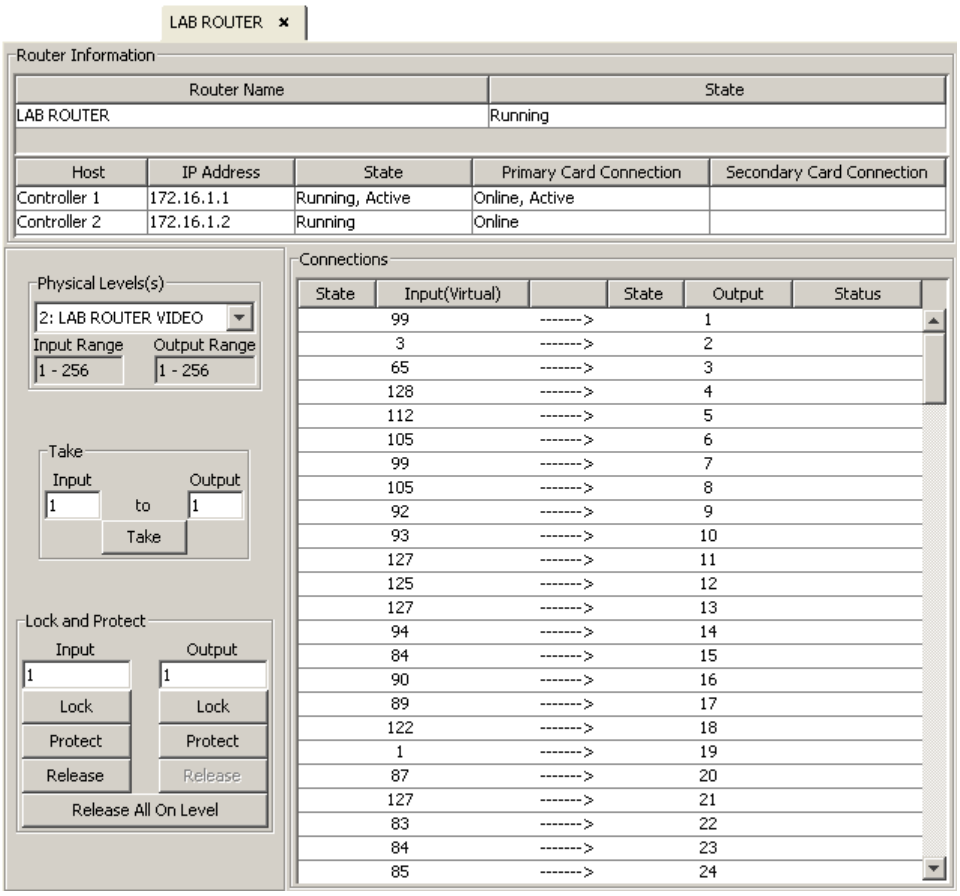
Virtual Crosspoints

At revision 4.6.12, the NV9000 software changed the way virtual crosspoints are handled. When you upgrade from early versions to version 4.6.12 or later, the NV9000 system controller loses its virtual crosspoint connections. The actual router matrices retain their *real* crosspoints connections and the configuration in SE remains unaffected, but the virtual crosspoints which were in the early 4.x NV9000 software are no longer in the upgraded NV9000 software.

You can reconnect your system's virtual crosspoints either manually or using a system snapshot. Once the virtual crosspoints are rebuilt, they are retained normally.

Manual Method

Obtain screenshots from the router views (under System Management) for all the physical levels of all the routers in your system. This is a sample:



Rebuild the virtual crosspoints from those screenshots.

It is important only to rebuild the virtual input to real output connections. The real inputs are intended to vary and it is pointless to rebuild any real input to virtual output connection.

Because there are probably several thousand crosspoints viewable, you could trim your set of screenshots to those that show only virtual input to real output connections.

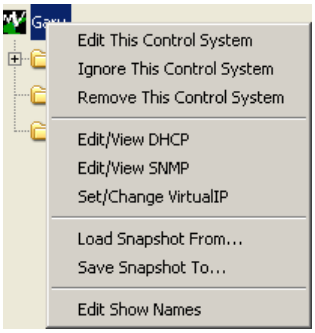
System Snapshot

Save a “system snapshot” from your existing system, then load that snapshot to your upgraded system.

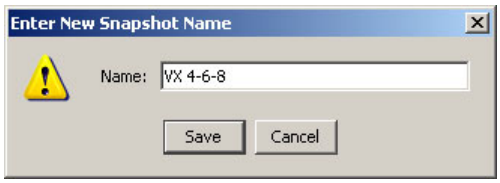
The following example upgrades a system with virtual crosspoints from 4.6.8 to 5.2.4.

Follow these steps:

1 Go to the ‘System Management’ page. Right-click the system you want to upgrade. The context menu appears:

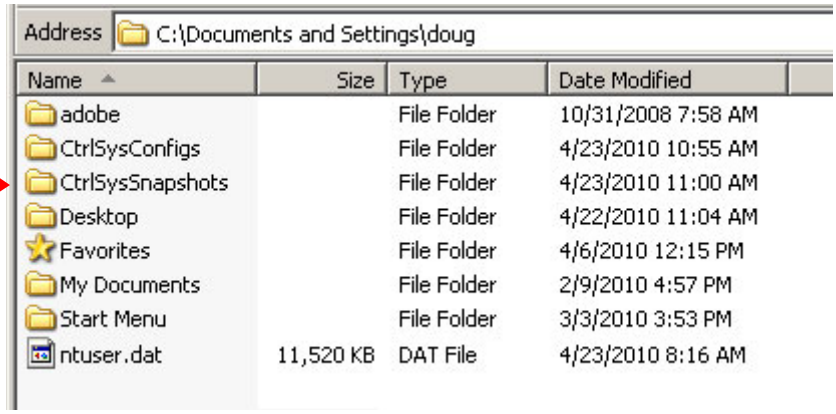


2 Click ‘Save Snapshot To ...’. A dialog appears:

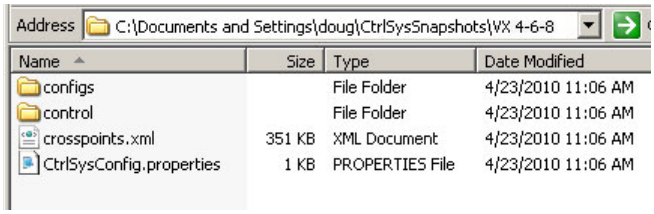


Enter the name for your snapshot. It might take a minute or more depending on the complexity of your system.

SE saves the snapshot in a folder in C:\documents and settings\‹user›\:



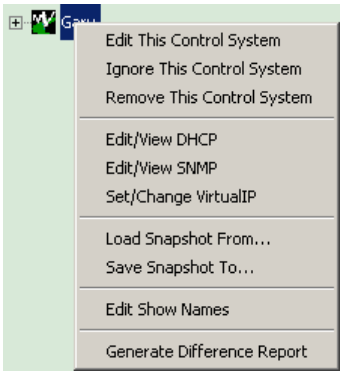
The system snapshot is a folder of the name you specified, containing these files:



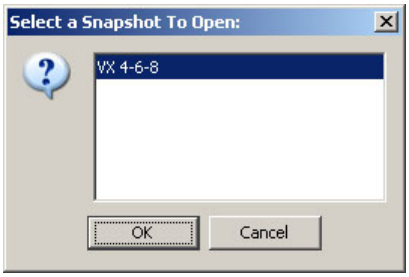
One of those files is a .xml file containing crosspoint records. We recommend you do not modify that file, but you may examine it.

3 (After upgrading your NV9000 system controller) launch your new version of NV9000-SE Utilities.

4 Go to the ‘System Management’ page. Right-click the system you have upgraded. The context menu appears. This is the context menu from SE 5.2.4:



5 Click ‘Load Snapshot From ...’. A dialog appears in which you can choose the snapshot to load:



Click ‘OK’ when you have selected a snapshot.

6 Wait for SE to load data from the snapshot.

At this point, the new NV9000 system should include all your old crosspoint connections, real and virtual.

(Loading the snapshot does not upgrade tielines.)