RUNDOWN ON 3D CRISS CROSS, PART II

A lecture given on
24 April 1962

Thank you.

Second lecture, April 24th, AD 12, St Hill Special Briefing Course and here we are on the particulars of 3D Criss Cross, continued from the earlier lecture.

3D Criss Cross is a process which addresses the Goals Problem Mass. The Goals Problem Mass is constituted of items, beingnesses that the person has been and has fought. And the anatomy of a problem is everywhere present in the Goals Problem Mass, meaning that you have item versus item. And these become suspended on the time track because they are in direct opposition, one to the other, and so tend to suspend themselves in time since there is no motion in the conflict.

So we have a situation where the beingnesses of the person have accumulated unto themselves, mass—accumulated in that series of lifetimes or that lifetime, and are actually composed of masses of energy. We formerly called these things "circuits" and we call them very properly "valences." But they aren't necessarily valences only in the Goals Problem Mass. Actual items would be valences, but these of course, have accumulated unto themselves things that he's been temporarily, people he has known momentarily—appended valences.

In other words, there are a bunch of other valences mixed up in the Goals Problem Mass. These other valences will never show up as items. Just the things he has been, the things he has fought, the things he has avoided will show up as items but these, of course—it's like so much taffy. And they accumulate to themselves all the other beingnesses associated. Any associated beingness to this has added itself to the Goals Problem Mass so that it becomes very large. And it becomes fantastically complex. There are probably tens of millions of valences and lock valences in a Goals Problem Mass. They are just innumerable. That's just a loose figure—that's to give you the impression of lots, you see. I don't think anybody will ever be able to count them.

Now, each valence is composed of a lifetime or a series of lifetimes and has complete within it all of the pictures and energy and standing wave accumulations of that lifetime.

Now as we run back over this lifetime, we find in this lifetime, a time track—at least some of us do—and it contains recorded experiences, conscious or unconscious, of this lifetime. Now, when a person has had this lifetime collapse on him, he's got a sort of a black ball which is all made up of this lifetime's facsimiles. And you have, in essence, the past valence condition occurring in this lifetime because all of his past 3D Criss Cross item valences are in that condition. They are totally collapsed. It's a collapsed time track— crunch!

Now, we've got a collapsed time track which is fighting a collapsed time track. Interesting, isn't it? And these things usually occur as a couple of black balls or some shape or color or size. They are opposed to each other. And in a Goals Problem Mass, we have the exact opposition for every item. Every item has its exact opposition. If you think of these items as a round, black ball with pockets of thought all through it and made out of energy, you've got a pretty accurate picture of it.

And if you look at the terminal-oppterm combination, you've got two black balls, one pushed against the other one. And they're hard fought. They are avoiding or attacking but anyhow they're enmeshed. And you've got two black balls. It's the basis of the universe—it's the figure two. So it takes two valences to make a package.

Now, to the outside of these black balls you have all the other valences that have accumulated, which makes it very interesting. And in view of the fact that the person may have been this identity for many lifetimes, you've got all of those lifetimes composed into one black ball and then appended to it, all the other little black balls that he has been in collision with in that series of lifetimes and we start getting a large number. And then, of course, the opposition has a whole series of little black balls like warts on the frog that are appended to that. And you've got your two big ones and then each one of them has a few hundred or a few thousand little ones. See. So, let's say that the figure was—the average figure was about five hundred—for auditing purposes you'll find out this is somewhere in the vicinity of correct. There'll be somewhere between two and seven hundred lock valences.

Now, oddly enough he has sometimes been these lock valences, see, as well as the main valence, but it just wasn't all that important and it simply became a lock on the other valence.

So we have every identity the person has ever been somehow or another mixed up in the Goals Problem Mass. In other words, we have these—oh, these big figures of identities you know—just tremendous numbers—and all the identities of the things he has fought are mixed up in the Goals Problem Mass and all the little identities that had been impinged upon are caught in that—these are all appended into the Goals Problem Mass. And when we finally get ourselves straightened out on this, why, we'll find out that at first look, it would just look like a big, black mass. Or it looked like a bunch of broken splinters or it just looked, you know—it was an unsolvable mess. It's pretty ghastly. It's like—it looks like a pile of coal that somebody has melted. It has that much identity.

Now, 3D Criss Cross takes this thing apart. The first thing it takes apart is by listing. And when you list, you're going to get the main item and you're going to get its locks. And the main item stays and becomes your 3D Criss Cross item and your locks, of course, blown, more or less, and they are part and parcel of the list. Do you follow this?

Man! Come up to present time!

All right. Now, I know these things squash you out of present time. Well, what you do, if you can get the idea of two spheres—let's get the idea of two black spheres. Let's take it down in its essentials and elements. Now, I'm not—I've given other lectures as to how they got to be two black spheres, but let's just take down pure mechanics. Let's take down two black spheres the size of bowling balls and push them into each other so that they are partially interlocked. Because they are held this way against each other, they become timeless. See, they have their own time track and all that but they actually are floating—they're capable of being in present time.

They haven't got any time in them, see, they appear to be timeless. We won't go into it any further than that they appear to be timeless and they float in time independent of time.

Now around each one of these are the lock valences. Now, one of these balls we will call the terminal and the other ball we will call the opposition terminal. The opposition terminal turns on sensation in the pc: grief, misemotion, sensation, coughing—that's right—all that kind of thing. Those are all turned on and are the symptoms of the oppterm, so called. That's one of the balls. You know.

And then the first ball we call the terminal and that will contain pain. What the pc experiences from that will be pain. Both of these balls contain coldness and heat. And one can get awfully cold bucking into a pair of these. They're very cold. And the—each one of these balls can be tremendously encrusted in these lock valences and they look like little balls, little black balls plastered around and impinged upon the two big ones, so we've got this tremendous number of warts. Now, those are lock valences and the opposition terminal can have its lock valences and the terminal can have its lock valences.

Now, the bulk of the charge, of course, is on the terminal and the opposition terminal and the reason the charges remain charged is because they're locked against each other. See, if a person had never opposed anything or had never been in opposition to anything anywhere on the track, then none of these valences would ever be impinged against any other of these valences and you wouldn't get a cross transfer of current.

Now, what are these balls made out of? They're standing waves of electricity which take the form in the mind of actual mass and they appear to have mass. It's measurable mass. And this carries certain potentials. One ball versus the other ball. And then all the lock balls against all of the main ones. Now, you've got that picture? There is—that is not an allegorical picture, by the way.

Now, if we take a hacksaw and saw one of these balls in half—if you could saw this much light-massed guck and junk—it would appear to have some holes in it and particularly in the center and that hole is full of thought. It's full of ideas which are encysted in the mass. These are the now-I-am-supposed-to's, these are behavior patterns. Here's the survival ideas—all of it compressed from the engrams of which this whole thing is composed. But it has a hole in it and that's got ideas in it. And then there's a whole lot of little pockets around through the thing like little air bubbles and each of those has got ideas in them—that's if we sawed one of them in half.

That's why a person can be inside of one of these things and keep chewing and getting ideas from it but nothing ever happens. See, he can chew around inside one of these things and get ideas, and get ideas, and get ideas, and get ideas, and get ideas but nothing ever happens. And he's not disturbing the mass or the balance between these two items. See, he's just getting the ideas out of them. Now of course, when the pc is being one of these things, he can't see it.

Now frankly, he is being the terminal and is ignoring the opposition terminal and that is exactly where the pc is located. But he doesn't know at the time you address this which of these things he is. He could be any of the locks—in his thinkingness—he could be any of the locks, he could be the terminal, he could be the opposition terminal or he could be any of the opposition terminal's locks. He could be something on the order of a thousand to fifteen hundred different identities—just on this one pair. He wouldn't know the difference, one to the next. Now remember though, he is being one of these things and that which ye be, ye can't view.

You have no point—viewpoint of those things you're in. So, oddly enough, he will have an idea of this but he will not have a view of this. He's much more likely to see the locks or he's much more likely to see the opposition terminal than he is to see the actual terminal. Now that is one package—one 3D Criss Cross package, which is the two opposed whole track valences and their accumulated lock valences—those additional identities. Of course, if we cut one of these little raisins in half, we are going to get the same picture as the big one. See, you cut one of the warts, you're going to get a hole in the middle of it full of thought, and so on.

It's by the business of living that this has occurred. It isn't that anybody has mocked these things up. It isn't anything that made these things. You go charging around through space getting yourself beautifully dislocated and being something, gorgeously, and attacking something and supporting something, protecting something and running ought-to-be's on other people and running overt-motivator sequences all over the place and having a ball. And around where you are at, during that lifetime you is going to get an accumulation of electric energy and that's all there is to that!

Now, if we, by some magic, were to be able to disintegrate this mass, strip by strip, we would find out it was composed of engrams and locks. You see? Let's take one of the little raisins now and let's start taking it apart and we're going to get pictures. They're going to be kind of lousy pictures, but we're going to get pictures—they're engrams. It's like one of these magician's things where you keep pulling pictures out of it, see—pictures and pictures and pictures. Of course, we take one of the big ones out of this mass and we start pulling pictures

out of that thing and whooah! See? This is going to go on and on, because they may be pictures of a hundred consecutive lifetimes. See here—would have all crunched down and become it.

See, it's not just one lifetime to one big mass, you see. There could be a hundred lifetimes eventually condensed themselves and in that hundred lifetimes you've accumulated even more lock valences. See. So the idea of the unit terminal and the unit opposition terminal and the unit locks for each, do not comprise the idea that that was one lifetime—that's what happens to you in one lifetime. No, fortunately, that's not the case.

All right, that's just one package. There are quite a few of these packages. I would be adventurous to give you an exact figure of how many there was in each bank. That's a complete package. Fortunately, the number of such packages is quite finite. It is not unlimited. That's very fortunate because it's quite tricky.

You get the exact forces of a terminal, exactly balanced against an opposition terminal, it would take some doing. So we haven't done this very often. You haven't gone on doing this every lifetime, you see, it just happened now and then. This, we're fortunate in. And so, for all the trillennia that one has lived, he hasn't accumulated too many of these things which are exactly balanced which are exact packages. But the ferocity and balance of these things is such as to give the whole aspect of timelessness and he's most likely to be associated with, plagued by or upset in or behaving according to such a setup—such a pair of spheres and masses.

And he's liable to think that's the end-all of everything, see, that's the end product to the lot. Concentration will be on that very heavily. Characteristics will be established by that very much. But actually, he's been a lot of things. Let's say a billion years existed between the last time he had a pair and this time he has a pair. See, there's been a billion years of free track. It's not balled up. It's not even crunched up particularly—doesn't even get picked up. Well, you go back and run whole track engrams and you're on free track and you're gorgeously missing the GPM. See, you can just run all the whole track you want to and your case isn't going to get anyplace. Be interesting! It'll alleviate some psychosomatics but it's in the lap of the gods whether you—when you're going to run into one of these GPM valences.

And the probability is, is while you are busy erasing whole track engrams, you're erasing them from the viewpoint of one of these masses. Quite interesting. You're being one of these masses all the time and you keep wondering where that odd somatic is coming from. That odd somatic has nothing to do with anything you're erasing. It has to do with the GPM terminal or oppterm. All of a sudden you'll hit a dizziness or a sensationness somewhere.

And where does this come from? And you could run engrams and run engrams and engrams and you'll still be dizzy. Why? Because you're sitting in the middle of the dizziness and you're erasing the engrams while being the dizziness. You see?

Now, this is actually quite an accurate mechanical picture I'm giving you of the Goals Problem Mass.

All right, let's take—just for a wild guess figure—let's take a hundred of these. I mean, a hundred complete units, you know—the two big ones and the little ones—let's take a hundred of these things and let's put them all in a pile and hit them with a pile driver and then move in on them from the side and so forth, and it would be kind of a ruddy mess, wouldn't it? And that's what you call a GPM.

Now, why is it called the Goals Problem Mass? Well, each one of these things had a purpose. It had something it wanted to do and one of the basic methods of locating these things was to find a goal the person had and by finding the goal the person had, one could then trace down one of these terminals. So, Routine 3 became a process of identifying one of these terminals by first finding the goal and then finding what would represent the goal and we—it would

turn out to be one of these terminals. This is quite remarkable. But of course it often turned out to be the opposition terminal. It wouldn't run. you weren't running the right item. The other was right alongside of it—so it had its limitations.

So I began to look around and study this a little bit more. We call it a problems mass to keep it plainly in view that a problem is postulate-counterpostulate—valence versus valence, beingness versus beingness. You see? There are two things. You can't have a problem without two masses or two things or two ideas, opposed. So we say problems just to keep this idea in mind and so that anything you know about the anatomy of problems—how they suspend themselves in time and all that sort of thing—applies to the GPM.

And then we call it mass because it is mass. It's actual mass; it's not imaginary mass. It's quite measurable. Wait till you run into one if you haven't. They're quite measurable mass. All right.

The difficulties of—auditors were having was in trying to find the goal and then trying to find the terminal that located the goal because if anything, that process is too good. That does its job much too well. And the pc goes slithering out of these things and into the mass and around and about it and so forth, in a very dizzying and dazed fashion. He is really upset by this because he's colliding with these things, one after the other. In other words, this is a hard entrance.

Now let's talk in the subject of entrances. How could you find out what these things were? Well, you could observe the pc and find out what he was doing and then figure out what would do that. That would be the lousiest method. Many auditors use this method. I want to point out to you that this is a method and that it is used. People use it on themselves and auditors use it on pcs. They find this pc galloping down the street on all fours barking every few leaps, see, and they say, "Ha-ha, that's obvious, we'll just run him on a dog, see, because it's pretty obvious that he's in a dog's valence." No, he wasn't in any dog's valence. He was being a frogman. It wasn't the right terminal, was it? In other words, that would be the lousiest method of approach because it's susceptible to the most inaccuracy.

Now, why do we have to be accurate? Well, there's tremendous, tremendous liability to running the pc as one of these lock valences because, of course, it is appended to this great big heavy valence and the amount of charge which you bleed off against the pc is rather fantastic, without doing anything. See, it's not going to unlock the real reason this thing is hung up in space. It is only going to unlock, to some slight degree, its impingement against other locks. And he can just chew on this and chew on this and chew on this. All right.

There is this method, as I've just mentioned, of the fellow and you see him going down the street on all fours, leaping along, and so forth. Well, very possibly, there is a dog someplace in it. In fact, it would be hard to find a pc who didn't have all valences—either as main valences or lock valences, you see? But every pc would have different main valences. See? They all have them all but you wouldn't have the picture of the main valences and that's the picture you've got to get. And you mustn't get the pc in one of these lock valences because it will blow off and won't read and that'll be that. It won't stay there, see? Because they're just—pc's really never been them to amount to anything They are not very important in his lifetime.

What you've got to find and what I had to do was find a lot of doors. I had to find doors which would let the pc straight into the Goals Problem Mass, but easily enough so that he didn't get his silly head knocked off, because it's very easy to get your silly head knocked off with a Goals Problem Mass.

Now, there are a number of ways you can open it. It was very difficult to get in by goals. It was very difficult to get in by goals terminals. You couldn't at all get in with a guess and an assignment. This is—all this is very difficult. You've been finding these for years, by the

way. you found them with Dynamic Assessment and you found them this way and you found them that way but, very often you landed on a lock, you see, that didn't audit.

Now, how did you—how can you reliably collide with one of these items? And that is 3D Criss Cross. Any method by which you could get a test item, which would then lead to a main item, would wind you up with this system on one or the other side of a dual-opposed beingness. Now, this is very, very tricky—it's very tricky. It's actually getting an entrance that the pc can be and do, at that time, that he can stand up to and that will blow charge.

Now, the methods of entrance of this are many but the best methods and the worst methods, alike, all consist of certain steps. And these steps are roughly: list, differentiate, null and check. Those are your basic steps. It doesn't matter how you go about this—you still list, differentiate, null and check. And each one of these items is precisely done. There's a precise way of doing each one of these items. "Item" is used carelessly there—doing what each one of these operations—a precise way of doing each one.

Now, these doors, these entrance points, could be many but the best entrance points are established by first establishing some test point by an assessment. So, you have such an assessment as the Dynamic Assessment. Let's make a Dynamic Assessment. Now, whatever—the old-time Dynamic Assessment. Now, whatever we get out of that is just a test item; it's not a 3D Criss Cross item. It won't be until we oppterm it, but when we oppterm it, we will find out—all of a sudden we'll land square in the Goals Problem Mass, see? Bang!

All right. Now, ways of finding these test items are as I said, many. The Dynamic Assessment method: quite valid. Pre-Have Scale: the Auxiliary Pre-Have Scale that was developed for Routine 3D is very successful. Now we can't say, however, that we wind up with a test item when we get that first list item. No, that's a 3D Criss Cross item. Let's see what a test item would be in a Prehav Assessment. We go down the line and find that our level is Control. The pc responds on Control. All right. Very good. We want to know "Who or what would control?" Or any way you wish to phrase it so that it makes sense to the pc. And of course, the test item is "What would control?" No, no, not the result of what would control—no, it's just those words, "What would control?" is your test item. See? You understand that?

It isn't the result of what you'd get of all this, you see? It is it, you see. See, you've done a Pre-Have Assessment and you wound up with Control and although you're going to list from "Who or what would control?" and so forth, you've already found a test item and that test item is "What would control?" "Underscore" (quote) (unquote). "That which would control" is your test item. Of course, that's not part of the Goals Problem Mass but that will lead you straight to the Goals Problem Mass because the pc rocked on it on the meter.

So, we're going to go charging down this line, by listing, and there's our first action. We're going to list. Now, what exactly happens in a list? Well, I'll give you that fast enough. We're going to list. All right. "Who or what would control?" Well, we can say, "Things," or, "You," or we can put it together—just so it makes sense to the pc, you see—this seems to be better to him and it seems to be better on the meter, too. It reads better when we say, "Who or what could control you?" It seems to read good. "Who or what would control things?" That doesn't seem to read so good. you get the idea? You kind of dabble this one out so that your test item there reads a bit. And then you say, "Who or what would (whatever your test item was)," see?

And we list, and we list, and we list and we list and as we list, this tone arm should be moving—ha-ha-ha. And that tone arm should be moving and that tone arm should be moving for every twenty minutes—optimum movement would be at least—I mean this is—this is very good movement: three-quarters of a TA division every twenty minutes. Ah, that's real hot movement. Poor movement is .25 of a TA division every twenty minutes. That's poor. Rotten, of course, is zero movement every twenty minutes. There's something real wrong there and so on. you actually, probably wouldn't play along with this 2.2—this .25. You'd probably start listing and you'd find out that's all that was happening and you'd say, "Skip it."

You'd do something else. you would ask the case some more questions as I will go into in a moment. And you would do some other things—chief of which is if the person has been run on 3D Criss Cross or Routine 3—any Routine 3-type process—you would complete lists and straighten this case up, but if the case has not been, you would of course, promptly put him on the CCHs and Prepchecking until you established tone arm motion. Okay? Those are the remedies.

But let's say we found Control as a Prehav Assessment, not to get involved with this, and we found this Control and we listed, "Who or what would control you?" and we wrote items. Well, that just consists of writing items. At the top of the page we put: "Number 1, period. Who or what would control you? date and pc's name." you will learn that when you haven't done this, you're going to have trouble because this list may someday be of considerable use.

And you just list. Well the best way of going about this—some pcs try to get economical and they like to give you only six or seven items—and very early on, if you're very ignorant indeed, you will take the six or seven items and you will put them on the page there and you will null them and then—well, this is the wrong way to go about it all. Now, you want—you want lots of items. And this is a paper-consuming activity. It's a paper-consuming activity.

Now, the auditor who writes—tries to write three columns on the same page eventually won't be able to find what he has crossed and what he hasn't and what he's marked and what he hasn't, and so on. And the best thing is just give up—just skip that terminal you have, "paper saver" and just go for broke and start right on down and list one column on the page. Then you turn the page over, and of course, you don't have to write all that data on the backside of it but you do have to write: "2—page 2" and you'd list on that. It's best to use legal length paper, 13 inch by 8 inch or 8 1/2 inch paper. All right. Now, we come to the next sheet and we write: "3. Who or what would—? date and pc's name."

Why do we do that? Well, they sometimes get separated, that's why, and we don't know what we're listing. See? So we make sure that every independent piece of paper on which we are doing a list has the name of the pc, the date, and in particular the exact question from which the list came. And we turn that one over and write: "4." (and we don't have to label that) and then we take the next sheet and we write: "5. Who or what would control you? John Doakes, 24 April 62." See?

And we go on this way and we finally wind up—we have wobbled the TA fresh out of motion, more or less. It's usually gone to a blowdown and the pc says it's on the list. We ask the pc, "Is it on the list?" But we ask this as though it's a repetitive question. Now this is important to you. I'll go over all these points.

You say, "Who or what would control you?" and the pc answers. Now sometimes he gives you three items without running down, you see, sometimes he gives you five without running down so don't keep interrupting him. See? Don't keep—and stop barking a Tone 40 acknowledgment. Don't you know the purpose of the acknowledgment is to shut the pc up and finish the cycle of action? Well what if you Tone 40 acknowledge every item he gives you? You've ended his list for him every time, haven't you? No, I'm afraid that a "good," is much better than "Good!" I'm afraid that sort of stops the flow.

Your idea is to get a list, see, not to stop the list. And it's not really a cycle of action of an auditing question that you're doing You're just helping him get the list. So everytime he stops talking or slows down, you shoot him the question again, "Who or what would—?" and so forth and you thank him. But actually, the fact that you're writing it down is acknowledging the hell out of it. He—it must be important, you must be getting it because he can see your pencil wiggling

All right. When we finish up this list, the pc is asked—well, it's slowing down, the pc says, "Well, that's all." First test is: "Well, is the item on it? Is 'Who or what would control you' on this list?"

And the pc says, "Well, I don't know."

And you say, "Well, come on, give me some more." you see? You don't monkey around with the meter or anything You bleed it down.

All right. And he eventually gets down to "Who or what would control you?"

Well he says, "Yeah."

Well, you ask him, finally, you ask him again, "Well, is the item on the list now?"

"Oh, yes. Yes, that's it."

Well, that's great. Now you turn over to the meter and you ask the question on the meter and see whether or not you get a rap. you ask him, "Are there any more items that belong on this list?" And if you get a rap, you first ask him, "Is there an ARC break?" See? That's modus operandi absolute. Make sure he doesn't have an ARC break, an invalidation or an out-rudiment because you can keep stretching a list on the ARC break of stretching the list. This is a big frailty that you run into. If you haven't got any fall on that, then there's always more items so you keep listing until your meter is null on the question, "Are there any more items on this list?" Okay? Make sure that that knock however, is not coming from an ARC break. All right?

Now your list is complete—possibly. It's a pretty good average that your list is complete at this stage. The pc has told you that it's on there, you've asked the meter, you've squared it up, you've added everything to it that you can. Now whether you number them or not, I couldn't care less. I don't care whether you number these things or not. It's all right if you do, it's all right if you don't, but there's no use for the numbers particularly, unless you want to cross-referral, of something or other. But there's no real purpose for the numbers.

Now, now comes your first actual test. You've asked the pc, you've asked the meter. Your first actual test is contained in the first twelve items. Of course, you make sure your rudiments are in before you start to null. But, look-a-here, the differentiation step is the next step which you do, but let me point out something to you. The differentiation step was done in an effort to minimize the number of items on the line and then tests demonstrated that when you asked the pc whether or not you should keep the item on the list, the pc very often crossed the item off the list because of the suppressor factor.

In other words, the item that you would have gotten on the list had been crossed off by the pc before you got a chance to null. This was too much. So differentiation would consist of, "Would (blank) control you?" (whatever the list item is). And the pc says, "Yes" or "No." And that's all. "Would (blank) control you . . ." Yes or no. See? You don't do anything with it. you just read it. you just ask the pc, staying in communication with him. That would be a differentiation of the whole list. Let me call to your attention something: that it becomes an optional step, because if the list is complete, there is no more charge to be bled off. Do you see?

There has been that alteration. Now we're going to keep differentiation in the lineup. And we're going to do it or not do it to the degree that the pc wanders and the amount of charge you can get off as a result of it. you want to go through a whole list and differentiate 450 items, why, fine. Ask him that question 450 times, that's fine, you might find it beneficial. But if you don't find it beneficial, you'd better not do it. You'd better just go straight to "nulling." Follow this?

So nobody's saying you've got to differentiate at this stage of the game. Nobody's saying this at all. It might look like no auditing to the pc. There have been many complaints from pcs about differentiation—ARC breaks because they were differentiating. They weren't getting

closer to finding the items. They had a limited number of auditing time and they thought it was useless. You understand?

But a pc who's getting muzzy and foggy and that sort of thing, you might find that it's a very good thing to differentiate the list in some fashion or other with the pc, but you might not find it's necessary to differentiate every list. Because all it is, is an orientation point. It blows a little more charge. Remember, that a complete list bleeds all the charge off anyhow. You follow this?

Audience: Yes.

All right.

Now, this has left you up in the air—Do I differentiate or don't I differentiate? Well, you'd better learn to make up your mind up about something I don't care if you differentiate or not differentiate. You'll find many a pc goes out of session during differentiation. And similarly, many a pc doesn't know what the hell you're talking about during nulling if you haven't differentiated. Furthermore, you might read the list to the pc back again and all of a sudden the pc thinks of a bunch more items.

You start differentiating and the pc thinks of more items, so differentiation can be used to extend your list. If the pc is being very balky, differentiation could be used as a mechanism to extend the list. If you did that, you would say, "Would (blank) control you?" You'd ask the pc and the pc says, "Well, really not." Slide one in about that time, "Well, is there anything that you haven't got on the list that would?" See? You make him think about the subject, pull him into the session more.

You could use it this way. I'd differentiate every pc that gave me twenty-five items and said, "That's it." And then I would use it very cunningly in exactly this fashion:

"Would catfish control you?"

And he'd say, "Well, yes, yes. Catfish would control you."

"Anything else there might? Any other item you haven't got on there that might?"

"Control me, control me—well, it was one of those—yeah, one of . . ."

"Any other item that might control you?" And so on. We coax him into that nice, long list. you could use this, you see. So differentiation has its uses. But I do not think it is a must. Okay?

Audience: Mm-hm.

All right. Our next item then is nulling. Now, if a pc tells you that it must go on the list, you put it on the list—during listing, differentiation or nulling At any time the pc says he wants an item on the list, you put it on the list. you understand? I don't care what it is, you put it on the list. And every time the pc says he wants an item off the list, you leave it on. That's all you need to know about types of items. You don't have to know anything about whether it's proper names, do they or do they not go on the list. It's an item, you put it on the list. Is it any other type of peculiar item, a present time item or anything like that, do you put it on the list? It's an item, you put it on the list. That answers all the questions. If the pc wants it on the list and if it is an item, you put it on the list. you see that? And there is no adjudication on the part of the auditor as to whether it is an item or not.

There aren't any items left off the list. If it has occurred on another list as "the item" and the pc wants it on this list, you put it on this list. Okay? In other words there aren't any items you keep off the list. you put everything on the pc wants on the list. That's it. Period.

There should be no question about that. But if the pc mentions something and doesn't know whether he wants it on the list or not you don't put it on the list until he tells you to put it on the list. That's also quite clearcut isn't it?

He says, "Well, I don't know—catfish, catfish uhh-hmmm. Catfish? And so on and so on and so forth—No I don't want that on the list—that wouldn't control anybody."

You haven't put it on the list yet, 90 therefore you cancel it. Your response to this sort of thing is to the pc, "Well, do you want that on the list?" And if the pc says "yes," you put it on the list and if the pc said, "no" you don't put it on the list. But once it's on the list, only the E-Meter and the auditor can take it off. Got it? See? It goes on the list quite indelibly. You won't have any trouble. Pcs are very adventurous in putting things on lists. Then they think. They almost never think before they put it on the list.

All right. Now, here's your first test of the completeness of a list. Take the first twelve items and null them. You'll find if you have—you shouldn't be repeating each item more than about seven times. See? You want to get three null reads in a row to call a thing null on that first list. Okay? That first twelve. And if you've got four or five alive, your list is not complete. The symbols which you use are an X and a slant.

If the item is null for three consecutive reads—catfish, catfish, catfish— you put an X. And if the item is alive, constantly or sporadically, but doesn't go null for three consecutive reads, you put a slant. And you put a slant for every time you test it. In other words, every time you come past that section, you put a slant. If you put an X down, you don't go back over it again except on the checkout step. So, here we go:

We say, "Catfish, catfish, catfish." Here's the way it ought to look: Catfish—read, catfish—read, catfish—didn't read, catfish—it didn't read, catfish—it didn't read. "Thank you. It is out," is what you say to the pc.

In other words, "Catfish," read. "Catfish," read. "Catfish," didn't read. "Catfish," didn't read. "Catfish," didn't read. "Thank you. It is out." Okay? All right. "Dog' next item. "Dog, dog, dog'—three reads. See? "Dog," not much of a read. "Dog, dog, dog," all three of them null. "Thank you. It is out." See, you're trying to get three nulls. See, you've been reading before, only against kicks. Well, this time you can do it three times on a null and you'll come up with a final result. See? In other words, you try to scrub those first twelve and if you can't scrub, oh I don't know, ten of them, you haven't got a complete list, that's all.

So you just put aside the nulling and you say to the pc, take up your pencil, look at it, square it all up, "All right, let's complete the list now." And he's so dismayed by all of this that he gives you items. Okay?

If four or five are alive, oh my, that list was not quite complete. If six are alive in the first twelve you read, oh man—that is nowhere near complete. If ten are alive out of the first twelve, you haven't begun. You just haven't begun to list, that's all. The whole thing is charged up.

Now, before nulling, always check your first two of the end rudiments, and invalidations, and missed withholds. In other words, let's get the half-truths, and the meter, and invalidations and missed withholds. Let's just check these items over and make sure that they are out. That's before you begin nulling in general. Well, let's make sure we got our rudiments in, in other words.

I should call that middle rudiments. But you can always tell whether or not your rudiments are in. Two things are going on if the rudiments are out: (1) all the items are flat or they're all firing equally. I think it will be a long day before you see them all fire equally but I have nevertheless, seen a pc actually so far out of session that every time you, the auditor, said

anything, you were charged, so therefore you got a fall on everything, equally. I have actually seen this, so beware. You'd get something like this: "Catfish," half-a-dial drop. ''Catfish,'' half-a-dial drop. "Catfish," half-a-dial drop. See? Routinely, right on the instant read. See? It's because you were speaking. But then you will see this.

There are ways to determine this and you needn't clutter your mind up with them but I'll tell you one just as an amusing thing You say—you say, "Catfish that can't swim." Let's say that's the item see, that you're testing, "Catfish that can't swim," see. Well, it falls on ''catfish,'' and "that," and "can't" and "swim." It falls on everything, see.

Now, if your rudiments—the other way is, if your rudiments are in, the first time you read an item, you're going to get a bark on it. There's some charge on it or it wouldn't be on the list. See?

It's when you don't get any read at all that you start worrying See, what you get is a reducing read. In other words, the first time you read it, you got a reaction, second time you read it you got some reaction, third time you read it you didn't get much of a reaction, fourth time you got none, none, none. you understand that? Your rudiments are obviously in. See, the thing is reading and then not reading. If your rudiments are out, you get a monotonous similarity of the meter that had nothing to do with anything you're doing. See? Usually null. So you got catfish, catfish, catfish, you see—that's null. Dog, dog, dog, that's null. See? Each one's null. you say, ''Catfish, catfish, catfish, that's null.'' "Dog, dog, dog, that's null." "Cat, cat, cat, that's null." "Room, room, room, that's null." "Light, light, light." Improper— improper needle behavior. See? Proper needle behavior goes: catfish, fall; catfish, fall; catfish, tick, no fall. See? You know your rudiments are in if you're getting any reaction. They're always instant read and they always fall immediately on the end of the item. Okay?

All right. You pursue that on down. Frankly, the way your list ought to look is one strike on the twelve—first twelve items and eleven Xs. That would be just dandy. That's very good. That means a very thoroughly discharged subject. And, now your next action—see, is getting very particular, we know how the meter looks on these things better. All right, your next action would be to go over the thirty-seven remaining Maybe we had 450 items, we covered them in one scrub. See, we went over it just once and we had thirtyseven left. All right, so we went over the thirty-seven, we wound up with five. We went over the five, wound up with two. Went over the two, wound up with one. The last two or three surrendering the most arduously. That would be a proper nulling See? You did 450, covered them once and had thirty-seven left. you do the thirty-seven, you wind up with five.

See? Thirty-two scrubbed on just one cross. You understand that's following the system of about no more than seven reads but with three consecutive nulls to be satisfied. Hm? You'll find that's quite a good system.

There's several systems involved with all this, by the way. I'm now using this system because you need a cross-check system for a complete list. And that's why we're departing and using this other system. I think you also— you'll find it a little faster.

All right. Supposing you did the first twelve items and you found that ten of those items were live and you couldn't scrub ten of them. Oh well, brother, you've got two out and ten alive. That list is incomplete and so we are now going to continue listing.

Now, we'll use the next twelve items as the first test. We'll use items thirteen—twenty-four, inclusive, and we found out that five of those items were alive and seven were null. Your list is not yet complete so we're going to complete the list further and now we're going to use twenty-five to thirty-six inclusive, as our next test. Got that?

We're sneaking up on the thing. In other words, we're winning all the time. We're getting more lists done while we test which is pretty clever. See? Always take a fresh sequence. You

don't go over the same ones you've gone over. To hell with those. Catch those the next time through. They'll be flat because your charge is bleeding off.

Now, what actually is this list composed of? The list is composed of: a terminal or an opposition terminal and all of its locks. It's one or the other, and all of its locks. Not all the locks of both, just the locks of one. And that list is more or less, a list of all the locks of this thing. And the individual, as he looks around, if he spots them all and the terminal itself, will blow all the charge out of the area and nothing is left in there to do anything.

Now, you oppterm—well, now you check. Let me tell you how to check. Way you check is you get your rudiments in. Get them in quite thoroughly and then read the item and then read every item that has kept banging earlier. You can tell them because they've probably got— they should have a slant and an X after them. Where everything else has an X, you've got thirtyseven items maybe in this list that have a slant and an X. So you read this item and the slant and the X item, see, and this item that you found and the slant and the X item—and this item you found has to keep on reading.

What happens if it stops reading? Huh-huh! Well, there's two things could be wrong Your rudiments have gone out and the pc has invalidated the item, one way or the other, and you have wrongly knocked out one of these— one slant, X items. And you'll find that the item is one or the other items or you missed or something goofed. But the usual thing is, is the list was not quite complete.

So you complete the list and you only bother to null and check out what you completed, but then you check it against everything that stayed live more than one pass through. You see how you check? You take the item, get the rudiments in very thoroughly and then you check the item against everything that was resistant on the list. And you can tell what was resistance because it's got more of your pencil marks after it. If you've got anything on the list that goes twenty-five pencil marks after it—oh no—my God, that list wasn't even vaguely complete. What are you doing? You know?

But the item itself, when you're checking it, you don't bother to keep slashing it. See? You just check it against these other items. That means your whole list is discharged and the one item is left. And that is what is necessary before you say, for sure you have an item. With rudiments in, that item remains, and there is no other live item on the list. Now, of course, there's no other live item in these ones that went out the first time but there might be some ticks or kicks left in some of these other items. And, if there's too much kicks left in these other items, your list, again, is incomplete. All right.

That list is composed of the names, assignments and generalities that the pc assigns to all these raisins and the big bowling ball. you are going to be left with the big bowling ball. And let me tell you something—the pc gets certain that it is on the list, he will tell you which one it is and he is always wrong Why is he always wrong? Because he is being the one that is right and although he has said it, he usually can't even remember saying it. He never selects it.

If the pc tells you it's it, and that is the item, and he knows that's the item, and that is it—if you were fool enough, without nulling, to take that as an item, then you'd wind that pc up in more hot water than you could easily cool off. Why? You see, the pc least sees that which he is most being. Of course, that tells you that the item is submerged. The pc knows about it, it isn't it. So, you get down to the end of the line and he says, "Well, it's on the list all right. It's that catfish. Ha-ha! Right there at the end. you know—I know that's it." you say, "Thank you very much." you appreciate the fact that he has told you the list is complete but you don't pay any attention to the fact he says it's a catfish because it's not a catfish. Not even vaguely. It is a fallen leaf that occurs eighteen from the end. Fallen leaf—and the usual reaction is, "Did I put that on the list? Is that on the list?" You know? It's that type of reaction. Now, do you see how to check them out? Hm?

Audience: Yes.

All right. Now, those are the entirety of the actions of 3D Criss Cross, except how do you patch up a case? Now, if you've got somebody who ambitiously has a long line plot already done—I've just learned this in the last twenty-four hours, this particular item—you immediately suspect that the earlier items found by a Routine 3 are not on the line plot and these have been omitted. So the first thing you do is find out if there are any earlier items found by any process. Okay? Any Routine 3-type process, goals, terminal, that sort of a thing—we're not too interested in earlier Dynamic Assessment items. See? We only want to know those Routine 3 items. See?

That's the first thing we suspect—that the line plot was started out with 3D or 3D Criss Cross and was not started out with the first career of the pc on a routine that headed him toward clearing. "Any item been found with a goal and terminal? Any item been found running to Clear? Any item been found? Any item that might be important? Has any item been found? All right." Then, we check the validity of the line plot. And you can put that down in letters of fire as your first action, is (1) check the validity of the line plot. I have seen line plots that didn't have a third of the pc's items on them. I have just seen a line plot within the last hour, which omitted the first four items of the pc.

Four items were found and then the person was run on Routine 3 because it now had a name. He was run on Routine 3, more or less, earlier than that. Four items were missing so the line plot is the first thing you suspect. Therefore, you make up the line plot as accurately as you possibly can. And any time you find an item that has been found and run in a Routine 3 manner on the pc—not now reacting—you discover the source of the list and complete that list. And that is everything You find the earliest item that was supposed to have been alive but wasn't, find out what list it came from, complete the list and find that item and then oppterm. You always oppterm.

This is the other rule with Routine 3D Criss Cross. This rule was sloppy. That's why these things were on Information Bulletins because they were not finalized. That is the rule. If you find an item, you oppterm it. In other words, any time you find an item, find the reverse item. you find "catfish" as a result of "Who or what would control you?" You finally find "catfish." All right, that's a Routine 3D Criss Cross item. All right. Now we want, "Who or what would oppose a catfish?" See, we list: "Who or what would oppose a catfish?" So, any pc that had four earlier items before somebody started running a line plot is missing eight items before the line plot even begins. Wow! Horrible, ain't it?

You'll find that any case will patch up and repair if you find the earliest item that wasn't right and then complete the list, and having completed the list, make sure that item is right and then oppterm it and the charge will bleed off that case the like of which you never saw before. Marvelous to behold.

If a case doesn't immediately come back to battery by this type of remedy, if it doesn't immediately come back to battery, with this type of remedy, and I do mean right now, just listing it, you see. Just in the session and listing it—oh no, there's something wrong with the line plot or you haven't got the earliest item and so forth. It only takes you about an hour and a half or two hours to find out if the case is going to come back to battery. Because you find the earliest one that was wrong and start making it right and the case will snap back.

Now the reason for that is the Goals Problem Mass is stacked, not in order or in consecutive sequence on the time track but in order of entrance by the pc and hell enter those corners most easily entered and it proceeds from that point. And you'll find there's a squashed whole package between something and something every time you've missed one. And it gets to be the muckiest mess you ever saw in your life. you found—you found this first item and it was all right and then you found another—didn't oppterm it—but you found a second item, and it was all wrong and then you found a third item over here. Well, already you've got a wrong

item against a missing oppterm. Well, that's enough to upset any case. The pc will be ARC breaky and so forth.

Now, what if a case has found a wrong item that won't checkout with you—you just check out all items the same way, you see. The checkout, when I say, "check out," I mean this operation of taking the item and checking it against the list. you understand. You'll find out that you don't have to do that ordinarily. You look at the list and it's got thirty-five slash marks after every other item. you say, "Oh, bluah," and just start listing again. You don't bother to go over these items that have already been nulled. Two things you can do with a list: You can suppress one or you can null one. And when there's twenty-five items—slash marks after every other item, you've suppressed items. You haven't nulled anything. You've just ground it in. you see. Takes much longer to do that, by the way, than to carry out a complete list. Because a complete list just nulls just like—oh, I don't know, it nulls like brushing down cobwebs. See, it's much, much faster to complete the list even though the list is terribly long than it is to try to grind out a short list.

All right, in the final thing—there's only, maybe, one other thing that you really have to learn before you develop some new mistakes is—one other thing you have to learn and that is: If a list has a wrong item on it, the line has a wrong item, and that wrong item has been opptermed, of course the pc has really been wound up in the soup because you've now done a whole oppterm list to a lock, you abandon that second item on the line.

In other words, it was "catfish," only—catfish is the right item, see, but he's got "dog." It's line A—dog, is right, see, but he had line A—catfish. And then somebody—"Who or what would oppose a catfish?" do you see, and he got a "fisherman." So we have line A reading, "catfish-fisherman." And we find out "catfish" won't check out so we just complete this list, whatever line A was (who or what would control you), we just complete that list, but we've got this item, "fisherman." What do we do with this item? Well, you just chuck it back in the bay, that's what you do with it. you don't do a thing with it. It will all erase and blow off and all will go gorgeously because the next line—it'll all disappear, there's no liability. It was hell—it was hell on the pc to oppterm a lock valence but this doesn't mean it won't blow off easily because now you just find the oppterm to a "dog" and you find out that's a "master" and the "fisherman" disappears and everything else blows. See? Everything is fine. you understand?

Audience: Mm-hm.

All right.

Now, I've told you very rapidly what amounts to, basically the essential operations here, of 3D Criss Cross and I have kept you a little bit overtime but I wanted to cover the whole thing.

Now, I have not told you how you do a Pre-Have Scale assessment. And I have not told you how the E-Meter is set to be accurate for Clear reads. Well, what I have told you is very surprise—surprisingly precise approach to 3D Criss Cross and frankly, you needn't add anything else into what I've told you. There isn't much else to be known that I haven't mentioned. It's terribly easy, but the first thing it requires is that your Model Session is good, that you can keep in these things and that you can read an E-Meter. If you can't do those things, you can't do 3D Criss Cross. That's all there is to it. Beyond that, those answers are precise. Everything you do from there on is precise. Okay?

Audience: Mm-hm.

All right. Well, I wanted to put it all down in one place, all in one gulp.

Thank you for bearing with me.

Good night.