Alright, so I think we'll get started. My name's Bruce Curran and I'd like to welcome you to this session. Between myself and Walter Bosch, my co-presenter here, we're going to hopefully give you a little information to make..give you some better understanding of DICOM and how it works and how to test it, and particularly we'll spend some time on the use of DICOM in radiotherapy so that this is a somewhat radiotherapy section..centric talk. I will note that if you have interests in it from an imaging point of view, there is an additional talk on DICOM tomorrow in the refresher course series that is a little more imaging centric. So I do want to acknowledge that I have borrowed a number of slides from a number of people, particularly from Sam Brain who he and I have been involved for many years in teaching a course as part of the Stanford University short course on IMRT and so he..I'm very thankful that I've been able to use

some of his slides. The Committee to Advance DICOM is a committee formed within DICOM and they have made a number of slide resources available to anyone who can pick up some Power Point slides that they're then able to use in their talks and I have taken advantage of theirs. In addition, DICOM Working Group 7 which is the working group within DICOM that specializes in extensions for radiation therapy, also has a number of slide resources and presentations that they have made available and that we have used in our talk..that I've used in my talk. So what are the objectives that we're going to try to accomplish in this talk? The first one is I'm going to try to give you a little bit of an introduction to what I'll call DICOM speak. As you'll see, there are a number of terms and strange acronyms that come up in DICOM that perhaps are not intuitive. I can say that having been involved in DICOM now for almost ten

years that there are still terms that I really don't have a clue what they mean, so I don't really expect you to get them all today but we'll hopefully give you some introduction to them. Part of that will be a little bit of the history of DICOM and a little bit of an introduction to the documentation that's available, where to find it and how to use it to your own best advantage. I'll talk some about DICOM testing. I'll tell you right up now I think what you'll notice most about my discussion on DICOM testing is how little material there is available for testing and some of the efforts that are underway to change that, in particular Walter Bosch will talk a little about some of the efforts of the advanced technology consortium to try to improve that state within the radiation therapy sphere of things. And then finally we'll spend a little bit of time talking about the DICOM connectathon. That's an effort that's being put forth by AAPM in

conjunction with DICOM Working Group 7 and with a great deal of assistance from the advanced technology consortium to try to push manufacturers to become more compliant with DICOM and its use in the radiation therapy world and to improve essentially the state of intercompatibility between vendors. So we'll talk a little bit about that and what we're trying to do in that project. So what is DICOM? DICOM stands for digital imaging communication in medicine. You'll often find it in lower case letters with a's thrown in and other things. But this is the official, if you will, definition of it. I should also say that DICOM started as a network or a communication protocol. It's designed for the transfer of information. It was not necessarily designed for the management of information. So that things like understanding when data is what's called stale data, when it's old, it's no longer valid, DICOM does not have mechanisms for that in general use. And in fact, Working Group 7 when it first started tried to do some management of data and essentially got their hands slapped by the overall DICOM management and the protocol standards group who said no, no, no, DICOM is not a management thing, you can't do that. Now they have since I think seen the error of their ways in that area and there are efforts underway right now to providing a more comprehensive way to manage data in something called work list, workflow in other areas and so that effort is just starting and I don't think we'll say too much about that today but there is efforts underway within Working Group 7 to provide mechanisms for this workflow management to take place. But it is not a part of DICOM as it was initially envisioned. DICOM is an object-oriented model, so for those of you

who are..have computer backgrounds that will make some sense to you. For those of you who are not, you'll be sitting there going yeh, so what? What it really means is is that in an object-oriented model you have objects which are essentially bundles of data and you essentially define your structure based on how you use these bundles of data. So you have a bundle of data and then you define operations that are handled on them. So for example, you might have numbers and you could consider addition to be an operation. So in DICOM we have an image and we have an operation which could be a store or a forward or things like that. And we'll talk a little bit about what that means. But essentially it tries to take a real world model and then abstract it into some set of objects and some set of processes that can be done on these objects, and we'll see that. The objects that are most commonly used in DICOM were originally images and with

the advent of radiation therapy extensions we've added things like radiation therapy plans, structure sets and other interesting objects. So this is sort of, they had a historical slide. This is essentially, on the very far side here is essentially what was originally called the ISO OSI model. It was a mouthful of acronyms. But essentially it was a model of how networking protocols should be written. And DICOM was originally set up in..not even DICOM but what was called the ACR-NEMA standard, it had its own physical connector, its own way of doing things and essentially it tried to take it from the point of view of the wire all the way to the workstation. When DICOM 3 came out, they essentially said well, we don't really need to do that. TCP/IP, which I hope most of you know, the inter..the protocol that's typically used in all Ethernet and other networking today, pretty much defines the wiring and a whole bunch of protocols for

making sure that your message comes in intact, whatever information is being transferred. And so DICOM de..DICOM decided that they could essentially work on that basis so they didn't have to do that. So when DICOM 3 came out, essentially they decided to say well, all of this networking information and all of this, this has already been standardized, we don't have to touch that anymore, and we can focus just on the protocol and how to transfer our own information. And so that in fact is how DICOM 3 is done. We're.. actually I should say DICOM 3.0 was the initial release and DICOM 3.2 is in ballot right now, so it has been evolving and continues to evolve with roughly a once-a-year update to it. So, as I said, it's an object data model. I've already talked about that. It includes a robust UID mechanism. A UID is essentially a unique identifier, it's a way that I can tell, I may have ten images, OK, ten CT

images that are all of the same thing but taken at different times or as part of a different series or studies. I may have a set of plans. Each plan, however, will have a unique identifier that should be unique throughout the world. Alright. So each manufacturer when they decide that they're going to do DICOM they apply to a standards body for their own, if you will, preface, and their preface tells..gives then a unique identifier so theoretically if you knew all these little numbering schemes you could look at an object and go oh, that was created by Impact on a such and such record and verify system model. Or was created by Phillips or by NOMOS or any vendor or by GE CT scanner. Because there is a unique identifier there and then the manufacturers are supposed to ensure that the suffix to that is unique within their equipment. And these are fairly long, I think they're 64 digits, so there's a lot of numbers there and so they ought to, with any

luck they ought to be able to keep things unique for a few million years if we're lucky. DICOM is really driven by a data dictionary. Really what it is is a mechanism where whenever a message comes in there's some identifier and the software goes and looks up that identifier and says oh, yes, this is the patient name, and I know something about the patient name, I know how the data's supposed to look, whether it's a string or a number or a binary or an image. So it has some information in the dictionary and so you can add things to DICOM very easily by just adding to the dictionary. And so really what the..a lot of the work of the working groups is really in defining new elements for the dictionary. Now there are in addition new services and other things that they add, but predominantly most of the DICOM work today is on trying to develop essentially new extensions to the dictionary to be able to transfer the information we want.

DICOM uses the term "service classes" to really mean operations. A service is essentially some operation like store, and as we'll see later this store will have some object that it's doing this service upon and these things become what's called a service object pair, or SOP. I'll say a little bit more abut it later but it really just says is is that here's something and I know how to do this to that something. So I know how to store a CT image. That does not mean I know how to store a nuclear medicine image or a radiation therapy plan, so when you define compatibility you have to define what objects you're going to work on as well as what services you can do on those specific objects. So for each object you would have your own list of services. And I think that's something that perhaps confuses us. A lot of us think well, it stores CT images, why wouldn't it store an MR image? But the answer is in DICOM is that there can be very different

requirements from one to another and there can..all applications don't handle all things. I've said already that it's a message protocol. I do want to emphasize however that there is a negotiation at the start of each transfer. Each DICOM entity that's sending information will then go to the application, the software application, the computer that is receiving the information and it will negotiate how it can transfer. Alright. This negotiation as we'll see includes, you know, can you accept a CT image? Will you not accept a CT image? It defines, there are data types. Do you accept monochrome 2? Alright. I'm not going to necessarily define that, do you accept monochrome 2 images or do you only accept monochrome 1 images? So there's a bunch of negotiation, there's compression in DICOM and so you have to negotiate will you accept a JPEG compressed image or do I need to give it to you uncompressed? So there's a bunch of

negotiation not only on what it will accept but how it will transfer it. In computer speak there's something called big-endian and little-endian which has to do with how bytes are ordered within computer words. Different computers do it different ways. And this has to be negotiated so the data comes across directly when they transmit binary information as opposed to text information. So there's a bunch of negotiation that happens to make sure that when the information comes across it comes across correctly. And at any point in this negotiation they could say oh, I don't do this. And the negotiation could fail. Alright. So at some point the systems may say we're incompatible, we can't talk to each other. Even though you think they could. And this is something that you don't see when you go to do, look at what's going on you look at a list and it says well, this one accepts all these attributes of all these elements of a CT image and here's my

receiver and it accepts them, this one sends it, they match, I ought to be able to send this information without any problem. But then you find there's something incompatible in the association and so it won't work that way. So there's a lot of stuff going on and it's not always easy to understand. So lots of different things. The data dictionary defines these different data types. It defines the different compressions. It also defines how things are communicated. How do I know, how can I indicate what I can support and what I can't support? It became very clear as things evolved that DICOM needed to not just be an online message protocol, which is what is to say, initial goal, but it needed to be able to do offline support as well. The model of freestanding imaging centers was the driving force for this where a patient would go somewhere down the street, get their CT scan or their MR scan and then show up at the hospital and say

well, here I am, and what do you mean you can't see my images or they didn't want to just have films. And so they developed a number of media, the original one was CD-ROM but now they've just released a standard for DVD disks so the images can be transferred and so that there is a standardization for how these images are stored, what kind of an index is put on the disk so that essentially anyone who supports media should be able..you should be able to take the disk from that freestanding center, bring it into your own department and if your application supports media, should be able to then pop in right in and it should just read without question. That's, of course, the ideal and in the words of some friends of mine when I worked in industry, of course it will just work. To make this all possible, DICOM has the concept of a conformance and a conformance statement. So theoretically everyone who writes an application that uses DICOM

is supposed to have a conformance statement. That conformance statement is supposed to state what they are able, willing to do, how they will..what objects they will accept, what objects they will send, how they will send them, what work they will do. And so it should be a means of testing and establishing compatibility. However, conformance statements in many cases don't go far enough. Originally it was just will you accept a CT image? And if you accepted a CT image it was pretty much given to you, accepted enough information to be able to at least display the CT image and it really didn't care much beyond that. As some of the extensions, particularly radiation therapy came into existence, the conformance statements became necessary to make them much more complete. We had to know well, would you send wedge information? Maybe you have a planning system that doesn't support wedges. But the linear accelerator really needs

to know whether there's a wedge present or not and they don't want to accept no information on a wedge as meaning no wedge. So some fields, suddenly we became field specific, data element specific, and what information we had to know was compatible between one element and another. And there are some DICOM applications out today that if they don't get all the information they feel they need they just essentially fail the transfer. And you're left with a situation as a user where suddenly your accelerator says DICOM transfer failed, or plan transfer failed, but you don't have a clue why. And so part of this is then going back and trying to figure out well, what data element is missing or was it an association problem, or what. So this becomes a very difficult issue. Some cases they may be that we want to push our vendors to be a little more communicatory in how they are want us to trans..how they want to indicate failure.

You know, there's nothing wrong with a DICOM association but I'm missing the wedge angle, or the presence or absence of a wedge. This is information that is not always well specified and we have to know. So DICOM terminology. There is something in DICOM called a DIMSE or a DICOM message service element. OK. Now I'm going to be very honest with you. DIMSE is a concept that is there but I don't find very useful in my day-to-day work. It's not something that I use and what I look at. But essentially it's supposed to be the set of DICOM application layer communications services. OK. What does that mean? What that means is is that DICOM has these layers like this old TCP/IP protocol and it's a set of the function, where the functionality is in DICOM, but as I said, I don't find that it very much applies to the clinical use of DICOM. It's there for the computer scientists and it makes sense for them but essentially if they don't have

that working for your application nothing will work at all. So once the application is working at least partially they've pretty much solved this issue. Where we start to get involved is in this information object definition. OK? This data abstraction that supposedly has enough information for us to be able to look at it and know yes, this is a CT image and it has everything I need to be able to use this CT image, or it has everything I need to be able to work with an RT plan or a structure set. So the information object definition is the key. The DIMSE and the DIMSE service groups are really infrastructure to the DICOM environment that is something that had to be there to make this work but personally I have not found a lot of issues in it. Walter may have something different to say about that but it's just not been something that I find very interesting, although you will hear DICOM experts within companies throw this around and

usually at that point that's where I say OK, I don't need to be at that level, I need to be somewhere else. Let's talk about something else here. Service class, however, is one of these concepts that's not very obvious but is something that is very important. The service class is the functionality, it's essentially what happens to the objects. What do I do? And you get to this service object pair. Alright. So when I put out a DICOM conformance statement I will say that I am capable of storing images. What images? CT images. What parts of a CT image do I keep? The Hounsfield unit conversion or do I throw that information away? Do I store it? Can I send it? And so you have to look at this and understand how you're doing, what you're doing. And we'll see how that works in a little more concrete detail here as we move on. So the whole point of the whole thing is is you have a data dictionary, you have some real world objects. DICOM

has molded this into an information object. They then added some services that they're willing to do, store, forward, and the result is something called an SOP or a service object pair, and you'll see that if you read DICOM objects for testing, looking at compatibility you'll see SOP classes and things thrown all over the standard, all sorts of elements there. What it really means is is a definition that this is the service, this is the object, this is what I'm saying I can do. Alright. It was..Sam Brain used to talk about this as a noun and a verb. OK. The object is the noun and the verb is whatever the service is. So the verb is the action, the service is the action you're going to perform on some object. There are a variety of service classes, verifications, storage, query retrieve. And then as you see, there are a bunch of management classes that are beginning to become more prominent. The one that we're most involved in is print. Many of

you have DICOM printers. I want to move on here or I will run out of time. I will say I have a lot more slides than I'm going to be able to talk about. I did that intentionally so that the handout, which will be available after the meeting, should have a lot of this information in it that you can download and have available to you. I apologize for not getting it in before. So service class, for example, there's a service class called ECHO. And the service class ECHO has an SCU, a service class user which is something that usually is somebody sending some information. It send a C-ECHO request to a service class provider, OK, which is typically a service and the service class provider should echo back. Well, for those of you who are familiar with trying to set up networks, this sort of sounds like ping. Alright. So service C-ECHO is essentially the ping of DICOM. It says, you know, is there a DICOM listener out there

somewhere who I want to talk to? And the listener sends back and says yes, I'm here. So it's really an identification, do we exist. And it sort of looks like this. So here's a typical thing where you have an SCU. OK. An SCU in this case is a CT scanner. The CT scanner has just produced an image and it wants to store that image somewhere. OK. So..or called a push. So there's a DICOM element service class called C-STORE, and so C-STORE says that the SCU starts this all by saying hey, I want to transfer this image to you, can you receive it? And if the service class provider says yes, I can receive CT images, then the images will come down and be stored. And there's also some, again there's some association that happens up here to start, and there's some verification that happens here on the end that says yes, I got it, everything came in correctly. And I've got a number of these, I'm going to just skip through them here, but there are

a number of these in the handout that you can see on different ways this can happen. So in the end we have, so here we see with printer, we see that it gets a little more complicated. We've got lots of things moving on here. We may have a printer server that does some conversion, we have some information the printer may not be able to handle, it may be a gray scale printer and we have color images so there may be some conversion that has to happen, so we also may recognize that the printer may not have the contrast or resolution of the image and so there has to be specification on how to manipulate this so that the image that's printed comes out as close as possible to the image that you saw on the screen. And DICOM has this functionality to do layout in gray scale control as well as queue management. I won't always tell you that we've set them up correctly but the management is there. OK. So here I am. I have a DICOM thing..a

computer come in and I have my, let's call it my treatment planning system, and I want to make it work. How do I set it up? Well, typically there are four elements that you have to configure for any DICOM application before you can start it. The first one is every DICOM application has something called an application entity title. It's really just an identifier and it says my name is George, and so that when you go to your CT scanner you'll tell your CT scanner that there's an application entity out there called George and it will know enough information to be able to send the data to it. So it's a name for a DICOM sender or a receiver. It's a maximum of 16 characters and by convention we tend to make these in all capitals. We have an IP address or a host name because I said we're using TCP/IP so we have to give it some internet network information. And then there's something called a port. A port is something we don't really know about but essentially TCP/IP has ports, it has various essentially addresses within the IP address so your IP address tells you what computer but in fact there can be I think it's 4000 different ports that you can establish a communication on. And different applications use different ports. Many people here know FTP. FTP has a specific port, when it does transfer it uses port whatever, I think it's….yeh. When you do Telnet, it has a specific port. DICOM has a port defined as a standard called Port 104 but in fact many applications will chose their own port and as long as the sender application and the receiver application both know what port you're using, you can easily set up this transfer. So these are four pieces of information that you need to make sure you have before you configure your DICOM entity. When you do that, essentially that information goes in and you can go through this model that I've already talked about. And

I'm going to skip this real quickly here. So in the end if we all got it together we've got something like this that really..that works. Now what's in a DICOM message element? So DICOM transfer is really just a transfer of a series of data elements. Each data element has a tag that's really an identifier. It is something called VR which is, may or may not be sent but it essentially tells me what kind of data is it, is it a string, is it a binary number. It has a value length, how many bytes are in this element. And then it has the data itself. And this is essentially a piece from a general study module so we see here are the tags. There's a DICOM, there's an attribute name. Each data element is called an attribute. So there's a unique name for it. There's the VR which is defined in part 5 of the DICOM standard which is the..part 5 says these are the data elements in DICOM. There are these tags, which are defined in part 6.

They're typically two 4-digit hexadecimal numbers, and the first number is called the group. And if the group number is even, that means it's a registered supported part of DICOM. But DICOM is also extensible so if the group number is odd, that means that this is a private group and you have to have a special application within DICOM to receive it. So a manufacturer is not constrained to wait. They can send other information than the standard using DICOM but there are special ways to label that and they label that by making this group number odd. OK. The tags are..there's a type associated with DICOM. This type is 1 if the element is required, has to be there, but it cannot be blank. So this means that the application always has to send this information and it always has to have valid information in it. A type 2 element is required. We always have to send the element but the element can be null, we don't have to actually put in any

information in there, we just have to..the receiver has to know that the element was actually in the stream. And a type 3 element is optional. And then in the text you'll find a free text description. Now the issue in radiation therapy comes in with these optional elements. These elements are often optional because DICOM, the radiotherapy extensions quite often are evolving with time. Like you might have an RT plan that starts at the simulator or CT sim, at that point it doesn't have beams, or maybe it has, but it might have a isocenter. So the beam information may be optional but it may be added later. But of course when it gets to the linear accelerator, if there's not beam information there, then it's pretty boring, there's not much, there's a problem. And so we have to worry about that. So in DICOM conformance statements this becomes much more of an issue. So DICOM RT started in 1980..94 at RSNA. And it essentially has evolved into these five major classes of objects, and because I'm running late on time I'm just going to quickly move on. But this is essentially the hierarchy, this is the, if you will, this is the DICOM model of the real world. And if you look in the standard you'll see this. So this is the way DICOM information is stored in the standard. You have a, what they call a module table. A module is just an easy way to group a bunch of attributes, a bunch of data elements into logical groups so that we can look at them essentially without having to look at pages and pages and pages of elements. And on this we'll see that their usage, some of them may be mandatory meaning that that group of attributes must always be present in an RT image. They may be U, which means that they're optional. And sometimes you will see C, which

means that they're conditional. So for example here, there's contrast bolus. Alright. If you use contrast bolus, it's mandatory that you include the information on it. But if you don't use it, it doesn't have to be there. So this is..this is how this is done. Now RT image is used to store simulator images, DRRs, portal images. It's essentially a way to take a radiother..a what I'll call a normal x-ray but a normal x-ray that has radiation therapy information associated with it. This is how the attribute information looks for one module, the RT series module, so you can see that we have information, we have it's type, and then we have some information on what's in there and what's legal. And in the handout you will find similar slides like this for each of the major DICOM elements in radiation therapy. In the beams module, the plan module, the major element is something called beams, and essentially this gives what we hope is everything you need to be

able to deliver a beam no matter whether it's a very simple static beam to a complex sliding window, IMRT treatment. And again, there's a whole lot of information here that I hope will be helpful to you but I'm going to move on, otherwise Walter won't have the ability to say anything. DICOM does use the IEC 1217 coordinate system. That's a defined coordinate system so even if your accelerator, for example, uses and I..uses very in-standard coordinates. When it goes into DICOM they must convert it to IEC and then the application at the end must convert it back, but DICOM always uses a one coordinate system, everything is always in millimeters or..yes, it's millimeters. So they have very standard coordinates and there are quite often been problems in many of the applications early on where people were using the accelerator standard conventions and not remembering to do the conversion. OK. So in the

DICOM standard there's, part 2 of DICOM standard is on conformance. It's a very short section, it's only about 20 pages, but you'll notice it's essentially the structure of a conformance statement but it does not specify a test suite. There is no authority within DICOM that says this is how you test things. It's really left up to the user at this point in time. And so I've left you a bunch of information here on how to use a conformance statement, but what I want to get to is the fact that I did go out on the web and I tried to look and say well, how can I figure out what's in these objects. And I want to just give you a brief introduction to two pieces of software. This is one called DICOMWorks. The http for how to find it is down here. It's freeware available that when you see here, all of a sudden we see that it will..allows me to take a CT image or an RT image and list all the information in it so that I can compare it to what's being sent if I think

there's a problem. DICOMWorks is very quick and easy to use, however you'll notice over here it says that it's 3004 and 300C are private. Those are actually radiotherapy groups but DICOMWorks dos not have support for radiotherapy data dictionary at this point in time. So it's very good for CT images, very easy to use. Then the next one I'll show you which is JDICOM but it does not support the radiotherapy extensions. JDICOM is a little more difficult to use but you see here that it very nicely gives me all of my radiotherapy tags for a dose object, tells me the tag it found, what the attribute is and gives me what was in the element so it gives me a very nice way to look at specific objects within the radiotherapy domain and it can be found down here. It has some licensing you have to go through with them but in fact, it does..they do a very nice job. So now I'm going to turn this over to Walter so that he can talk for a little bit about the role of the APC in trying to use DICOM.