Hello, my name is Greg McDonald. Today we are going to talk about upgrade managers role in your Cognos deployments. Hopefully by the end of this session, I would have shown you how upgrade manager can be used in the lifecycle of your Cognos applications. A little bit about myself before we begin. I have been at Cognos for just over nine years. I was in that support organization recently moving over to product management. Currently I am the product manager responsible for application life cycle and of course upgrade manager. Our agenda today, we are going to start by talking about upgrades in general, questions around upgrades and then we are going to take a look at the upgrade work flow. We will then start to discuss upgrade manager and work and help with in your upgrade process. We will discuss details and how it works, the type of comparisons that are available today. We will look at the *typology [Phonetic]*, supported environments and then I will do about a 15 minute demonstration. Time permitting, I would like to quickly discuss application life cycle and upgrade manager and how the two can be used outside of just an upgrade. I then like to show you four or five things that we currently have cooking in the labs for the future for upgrade manager. So lets talk about the upgrade.

Starting with the reasons that customers generally upgrade, align of business has requirements that can only be addressed using features and functionality of a new release perhaps since an existing line of business or a group of users who require new features and functionality that aren't currently available in their release. We also have IT, IT is making an architectural or a hardware upgrade in your environment were possibly upgrading Other software that inadvertently affects the version of Cognos software you are on. And then of course we have customers who just like to stay current. Now some of the challenges that I hear when you are upgrading, the first is the process. The process to IT and to the business. IT needs to understand what the process is and the business needs to know how long it's going to take. A lengthy upgrade costs both IT and the business. You have to understand how the work flow works. You are going to have to start evaluating reports. You are going to have to possibly fix reports and validate content. Now these are the types of questions I hear most often. So let's take a quick look at a few of the report upgrade problems. The main one being report validation between versions. Currently it's a manual process which obviously is very time consuming. There is many unknowns for example the #1 question is will my reports return the same results in the same amount of time and will the formatting remain the same. Then we have to start dealing with what are we going to test? Are we going to test a sample of our reports? Are we going to take a percentage of the reports from each line of business or are we just going to test 100% of our reports to make sure that they respond and display exactly as they did in our previous version. And then of course we have prompted reports. Journey upgrade cycle you may have consulting, you may have people who aren't familiar with the prompt values required to return results and this can be problematic. So we have taken a look at why you have to upgrade some of the Questions you have to answer during the upgrade. Now let's just take a quick look at the upgrade workflow. What we have to do is create an upgrade plan, the first step in the plan is going to be to review the documentation. Then we are going to have to conduct a site survey. That is going to mean contacting users and using auditing to determine which contents you are going to migrate or upgrade from your previous version. After that's collected, we will then create the upgrade plan. Now the first step in the plan is obviously going to be the create the test environment. That's why you will install and configure Cognos 84. You are going to then want to deploy the applications from your previous version to the 84 version. Once they are deployed and upgraded you then have the ability to test the reports. After the testing process you are going to have identified issues with the reports. That's when we fix them, you would then going to have to retest those fixed reports to make sure they validate against the previous version. At which time you may revise the upgrade plan.

Next step. We are going to have to prepare the QA and production environments for deployment. We will take our recently upgraded 84 test deployment and move them in to QA and the production. Once they are in QA and production you may then want to validate either against the previous version or the same version but in test where you originally validate it. I have left the three circles in red on the screen because these are areas where upgrade manager can quickly and easily help you validate the content of your reports is accurate. So lets move to an introduction of upgrade manager. What does it do? It's going to help manage the validation of report content by providing a work flow. Upgrade manager will quickly and visually that your 84 reports, return identical output to what they did in the previous version. Now we do that by validating run output. We don't look to saved output. The output types that we validate currently are pdf only. The types of reports or reports generated from report studio, query studio or analysis studio. Upgrade manager is configurable to many configurations and environments. It allows you to create many projects and each project can be configured to any combination of systems, whether they are in the different version or the same version. Most of today will be discussing upgrading, therefore discussing two versions. The functionality will allow you to easily benchmark, validate, revalidate report content when anything in your environment changes that may affect your reports. In a nutshell upgrade manager will reduce the time it takes to validate large volumes of report during an upgrade process.

How upgrade manager works? We query the source system for content. You choose which packages and folders have the content that you wish to validate. Currently we only validate content that's in public folders. Now upgrade manager needs to run reports, therefore it needs to be able to run reports without user interaction because it can potentially run hundreds of reports for you. We have the ability to generate prompts automatically if they are based on a database query. We can generate Prompts manually or if you need the prompt to be run, you can run the prompt and choose the prompt value. Reports are run once on the source and target system. This is important to note because during peak times or during times when your developers or user base is using the systems you want to be vary of how many reports you are kicking off. Upgrade manager retrieves all output locally. We do not write anything to reporting that or Cognos 8 content stores. Projects are saved with all configuration values for future validations. So the upgrade manager work flow. We start with creating a project, once the project is created, we enter our source and target system parameters. We then generate a report list. We do this by connecting to the source system, choosing the packages and folders that contain a report content. We are then going to move to the validation and execution in the source environment. It's at this point that you can exclude report content and you can also generate prompt values where required. We then move to the validation execution and the target environment. All prompt values entered in the source validation are carried forward. We then move to comparison and analysis, now which time we may or may not identify reports with differences. You then fix or approve the differences and then upgrade manager gives you the ability to re-execute in the target environment to compare against the source once again.

Now the type of pdf comparison options that we currently do is the built in comparison which is the first or automatic comparison. This comparison will simply tell you if there is a difference or not. If there is a difference, we offer two compared tools, a basic tool which will give you a side by side pdf version and allow you to visually compare the two pdfs. You do not have simultaneous scroll, which can make it difficult to quickly and easily identify where the difference may be. Now we do offer a text or data compare where differences are highlighted in red. We also offer the pdf flash output compared. This two offers are side by side compared, but with a single scroll, it also offers an overly compared allowing the system to quote one pdf over the other and give focus to either which will help you identify slight report differences. It also offers the text or data compare. Now upgrade manager will allow you to archive a test case for customer support. When ever you find a difference in a report that you cannot resolve or understand why there is a difference, you can use this feature. It does need to be enabled and it is going to create a zip file from both the source of target environment. The zip file can then be submitted to customer support. Some of the files that the archive function creates are the pdf output from both the source and target allowing any analyst to see the differences for themselves. The report specification, the sequel statements generate in both source and target, a model spec, the complete logs directory from both systems and then of course most importantly the versions, the component versions on each system. So the task summary and stats page. We will take a quick glance at this page to see what it will offer you. Now we give you the total number of tasks in the validation and execution sections. We also offer you stats on how your comparisons are doing. Whether there is differences or not, number of report objects that have been marked out of scope, whether a prompt is missing. We also give you the ability through out the work flow to add notes. These notes are all carried forth to the task summary page so that somebody else coming in to this system can see why there is a failure, why something has been approved or what a report object has been marked out of scope. We also have the total progress.

Supported environments. Upgrade managers are windows only tool. As a source, it supports report net 11MR3 and above and as at source and target, it supports 8283 or Cognos 84. So source and target operating systems, we support Windows, any flavor of Unix or Linux and upgrade manager allows from multiple projects with multiple project configurations. The Typology. Upgrade manager is a stand alone application, the browser based UI and requires access to both the source and target systems. The quick look at the typology. On the left side we have our report net environment. On the right side we have our Cognos 8 environment. Using the built in functionality you are going to deploy and upgrade the content. We then have upgrade manager in the middle which connects the both report net and Cognos aid validates and runs reports. It then generates the pdf output, brings it locally to the upgrade manager machine at which we will tell you whether there is differences or not. and I would like to now quickly move to a demo. Okay, so lets take a quick look at the upgrade manager splash page. We have our basic help about and help contents where you can get full access to the documentation. We have settings and we have new projects. Now on the bottom left hand corner of the screen, you will see two icons that are *Xout[Phonetic]*. This is an indication as to whether the source and target systems are available for access by upgrade manager. So let's create a new project. Now when creating a new project you have three options.

Create a blank project, create a project using an existing configuration or create a project from a copy of an existing project. The differences are when creating a project from an existing configuration, all you will save and use are the configuration parameters. Not the report content, we will choose that for this demo. Once it's created we need to configure the project. We select the configuration and you see here three tabs basics, security and advanced. This is where we add the source and target parameters. Upgrade manager needs to know the gateway and dispatcher URI of both systems and of course it needs to know the version of the source and the target systems. We then require the credentials to run reports in both the source and target systems. If you chose to leave these blank, upgrade manager would prompt you every time a report is run. And then we have some advance settings. This is where you would enable the use of flash, set the validation level and other basic parameters. Now once you have set the configuration, you save it, at which time, the generate report list option will be enabled. Now as you can see in the bottom left, the source and target system indicators are now available if I was to click one of them, you will see I will launch that version. Now we generate our report list. We are connecting to this source system, at this point in time, we will select which content, which packages and folders contain the report content that we need to validate. So once upgrade manager has validated the content, it brings and in it's available for us to start using the data now. We start with the tasks, source validate, source execute, target execute and validate and then of course the task summary. So the work flow is to start with the source validate. This is where we are going to make sure all prompt values are added. We are going to potentially mark things out of scope. You can see upgrade manager gives you a, an indication as to how many items are in each folder or package, the status of there something missing. Lets take a look in to the first folder. We have four reports, we can see all reports require prompt values. We are going to mark a couple of these reports out of scope. Now you can apply it all these actions to all source and target validations and executions. So you no longer need to retype information and then of course we give you the ability to add a note so that you can indicate anybody else using upgrade manager, why this report has been marked out of scope. In this case we are going to say that the report is not being upgraded. Now we need to generate prompt values for the two reports we do is to validate. Upgrade manager gives us the ability to run the prompt and select values. We can see the parameters missing and you also have the ability to clear all prompt values if you wish to remove the captured prompt value and choosing the other one. So we select our prompt value, click finish. We now hit the back button and move to our next report. Again run the prompt, type in your value in this case, click okay, the value has been updated. We are going to hit our back button and again you can see the status is now new. We will now move to our next package. All these reports have a status of new which means nothing is missing. No prompt values are required. In this case we are going to validate all these reports have no errors in the source environment. The status has been updated from new to valid, the last folder. Upgrade manager also gives you the ability if the prompt is based on a database query to automatically capture the values for you. So hit the drop down automatically prompt the value, and you can see the status has been updated from prompt value required to new and if you look in to the query parameter, you can see what parameter has been updated. So now we are going to move to the source execution. We are going to want to make sure that all of these reports execute in the source environment. First select them and execute. Status is succeeded, we had now gathered, the pdf document required for the comparison, select all reports and execute. Status succeeded in all accounts. Now in upgrade manager you don't have to drill in to the package or folder. You can simply click on the folder option and hit execute. So once it's completed executing, we update the status to complete it. Now we want to move to the target validation. Now we drill in to the first package and you can see there is no prompt values required because they have been carried forward from our source validation. You can also see that out of scope and then notes associated have also been carried forward. We will validate the two reports In the target system and with the target validation of this package you can see that we had one valid report and one report failed. So we want to determine why this report failed. We go to the properties of the report and you can see we have displayed the aero message from the Cog server log that the report server is not responding. If we click on the icon in the arrow, we are going to launch you right in to the Cognos knowledge base. At this point you can investigate different documents and resolve your problem. Once you fix the problem, you can mingle back and have a upgrade manager quickly validate the report is fine. The status valid, we can end quickly with this package move to the execution Now execute and generate our pdf output. We must do this for all packages. Now you can see in this stuff that I haven't validated. The validation step in both source and target systems isn't required, how ever, you may want to do it in some cases for quicker and easier debugging. And you can validate at any stage. We will finish by executing the last report, we have succeeded, we can see the status has been updated to complete it, which means all executions and pdf output has been gathered and just for the sake of argument we are going to go back and quickly do a target validate to complete the process, source of validation, execution, target validation and target execution have now been completed. We could now move to the compare. Let's work with our first package. We have two reports in this package to compare. We are going to select the reports, make sure the drop down is on compared reports and hit go. Now you can see there are no differences, so we can go to our next folder. Our six reports to compare. Again this is the automatic repair, the default compared which just alerts us to whether there is a difference or no differences. You can see in this example we have two reports with differences. Now is where we are going to wanted drilling and see what the problems are. In this case we will launch the default compared tool. Now you can see one comparing reports is not always easy to identify what the difference may or may not be. The reports look fun, but the tool has told us there is a problem. So what we are going to do is view the text or the data. Now you can see in here what's happened is a column has shifted to the right and therefore cause upgrade manager to alert us to a difference. In this case we are going to approve it and we are going to indicate that the approval is because there are all formatting change that makes … we are going to approve it indicating that it's a slight formatting issue that's acceptable for upgrade. We will now go to our second report. Again we will launch the tool and again the reports Looks slightly different, the columns all match. It's not easy to identify where the problem is. So again we are going to the view text compare, where we can see differences. Now as you can see I have updated the target database to indicate or to change a value. In this case we are going to reject the change indicating that further analysis is required.

Lets move to our last example. Again after hitting the automatic compare, we see there is a difference between the source and target pdf output. Go to the properties, this time we are going to use the flash output compared tool. Now in this case, you will notice you have the side by side, but with the radial buttons at the top for your scrolling. Again we can't easily identify where the change is, but in the flash compare, we have the option to do the overlay. Once we have chosen the overlay option, we have the focus option on the right hand side. This can give focus to either pdf, the target outsource. Again in this case we can easily identify where the upgrade difference is. Our best option again is the data or text compared. And now we can see we have an ordering issue. So again we are going to reject this and indicate with a note that it requires further investigation. All our comparisons are completed and we now move to the task summary. This is the overview of how the project is doing. We can see how items have done in the source and target, validation, source and target execution. We can also see that all notes have been carried forth to this page, so we can see why items are rejected, why items are marked out of scope And of course we can see the overall progress of the project. So lets take a look at the upgrade manager summary. What did we see? We have seen that upgrade manager can help journey upgrade process by validating the report content is identical a source and target system. It's easily configurable to many different environments, it quickly validates large volumes of reports including prompted reports. It allows you to easily retest once issues are identified and fixed. You provide a task summary and statistics page to quickly let you identify how the project is progressing and we allow for easy test case collection for customer support using the archival functionality. Okay, so lets just take a quick look at application lifecycle and upgrade manager. As you may have seen I base most of comments around upgrade. However, upgrade manager doesn't need the source and target systems to be a different versions. They can be the same version, so it's not just a tool to be used in an upgrade situation. It can help you quickly validate reports between environments, between *Delve [Phonetic]*, QA, and production. This will allow you to quickly identify problem reports or quickly move reports

From one environment to the next. This allows for quick benchmark.

Now application life cycle. Lets take a look at how our application lives and grows in your environment. We start with our development environment in the current version. We develop frame work manager models and reports. We progress those to QA and then production. This life cycle continues and is ongoing, however, we at Cognos are providing service packs and fixed packs including hot sites to resolve issues and provide new features and functionality. Along with that we are creating new versions of the product. Our goal is to get you from production to the next version as quickly and seamlessly as possible. Then once you are in the next version, getting you as quickly and seamlessly as possible from Delve to QA and production. At each step along the way upgrade manager can quickly help you validate that your content has not changed due to version, service pack, fixed pack or hot site. So let's take a look at what's in the labs. We are currently looking at all supported output types, Excel, HTML, XML, My folder content, so that not only can content that's in public folders be available to be validated in upgrade manager, but so can my content. Multi-lingual reports, we will allow for content packages and folders to be in different locations in the source and target system. We are also looking to remove the manual configuration. Now one of the big things we are looking at is the ability to have a single environment project type. So this would allow you to truly benchmark your Delve, QA or production environment than when something changes whether that's with our software or software that affects Cognos reporting. You can quickly and easily validate that those changes have not affected your report content.

Well, thank you very much for your time. If you have any questions for the upgrade manager, please contact me through my email address, Greg.mcdonald@ca.ibm.com.