

# Upgrading to Cognos Analytics

Every process can be broken down to a methodology. After having help many customers upgrade to Cognos Analytics, I have established a repeatable process. As anecdotal as it may sound, I will include some specific facts that may help you in your journey. The steps include

1. Planning the Upgrade
2. Environment Setup
3. Establishing a Baseline
4. Content Migration
5. Validation and Remediation
6. Go Live

## Planning the Upgrade

### What to Migrate

For many years there have been two basic approaches to migrate content from a previous release.

1. Lift & Shift
2. Consolidate & Optimize

In order to determine what is the best approach for your environment you really need to consider what is important to your users. The Lift & Shift method implies that everything is important and everything needs to be migrated. If the number of reports is relatively small, then including everything is likely a reasonable approach. However, if you have a large number of reports then perhaps a Consolidate & Optimize approach is more likely to succeed. Chances are that if you have used Cognos for a long time it will be more likely to have duplicates and possible opportunities to consolidate. Additionally, there may be some content that is similar and with better understanding of requirements or by using some of the advanced capabilities of the authoring tool some reports can be optimized. The challenge with Compress & Optimize is the time to deliver and the resources required. Perhaps a Hybrid or Batch approach is more suitable. In other words, maybe migrating contents in small groups makes more sense for your situation. This would give you the opportunity to control the schedule and resources required.

There are tools available that can inventory your environment and provide some level of detail to help make these decisions. Important factors to consider are the total number of reports and complexity of those reports. Using a complexity rating of all

your reports can help guide possible remediation estimates. Some of these tools can also help identify duplicates which can in some cases lead to significant reduction in report count. These tools include the Audit Report, Cognos Toolkit which is available from Cognos AVP or Expert Services and MotioCI.

Identifying what needs to be moved is the key step in the planning phase. Confirm that the reports are needed. Confirm the reports still work, if not fix them in the source environment (before migration). Consider archiving reports that are not defined as required for migration, that way the asset can be recovered should it be discovered after the migration as still needed. Limiting the scope to a manageable migration will help you succeed.

### **Testing Strategy**

After you have defined the list of reports that will be migrated you need to determine what needs to be tested. Do you need to test all of your reports or is there a sample set that can certify the upgrade? Engage your users to define their level of comfort in the migration process.

What scenarios need to be tested? Are there multiple use cases for a report that need to be tested? Do these reports have multiple parameters that need to be verified. Are there security requirements that would require multiple test cases?

Define the test cases. Collecting the parameters required can be time consuming. Some testing tools can collect parameters from your environment to automatically build these test cases. Both Cognos Life Cycle Manager (LCM) and MotioCI as an example will gather report parameters from previous executions. Make sure your environment has either saved output or run history in the audit database to automate this process. If those parameters cannot be collected automatically you may need to manually add them to the testing tools test cases or create report views in your environment to manage the testing.

How long will the testing take and what is that impact on your testing? Most environment will struggle to freeze all report development during this upgrade process. To address this you can either migrate and test in batches or manage the changes. Your governance process might help mitigate changes during the migration or if you have installed a tool like MotioCI it can also monitor for “stale” reports, reports that have changed during the testing cycle.

Testing tools as of this writing only support Reports. Cognos Analytics has consolidated the studios to dashboards and reports, all others are considered Legacy Studios. Of course, there are new components within the Cognos Analytics environment such as Exploration, Stories and Notebooks but those are not a consideration for this writing as we are addressing a migration from pre-Cognos Analytics (Cognos 11.x).

## **Other Things to Consider**

Factors that complicate a migration include making multiple changes in one pass. Some of those include:

- Migration to Cloud
- Migrating from CQM to DQM
- Changing the OS of the environment
- Changing the Interactive Flag
- Changing the Content Store database vendor
- Changing the database vendor of a datasource
- 

The key message here is to keep the migration simple and get to Cognos Analytics first then evaluate other changes. After having migrated many customers to CA the success rate has been very high averaging less than 1% remediation if the test plan is kept simple.

# Environment Setup

## Versions

Which version of the product should be installed? New with Cognos Analytics is the release cycle. You may have noticed that there is a cadence of releases several times a year which would allow for new features to be introduced and bug fixes. The impact of this rapid release is the support model which is best described as n-1 & LTS. The concept is that the latest release plus the previous release are actively supported. Additionally, there is a release that will be supported for a longer period, known as Long Term Support. In the case of 11.0 the LTS release is 11.0.13, details can be found [here](#). What becomes more evident now is that a good regression test strategy is essential. The recommendation is to stay as current as possible so your environment is fully support and for production you should minimally be using the LTS release.

## Considerations for the install

If you are familiar with installing, configuring and tuning older versions of Cognos you may be familiar with the general tasks but there are a few things that are different with Cognos Analytics.

One of the changes in the install process is the use of repository files. In previous versions of Cognos there were large install files, these are now separated into the install application and the files for either server or client. Details can be found [here](#). Cognos Analytics is leveraging DQM more than previous versions of Cognos and for that you need to include the JDBC drivers. Most notably is a new location for the database drivers, referring to the [documentation](#) the JDBC files now need to be placed in `<install_location>/drivers`. Additionally, if you have framework manager packages that have been deployed using CQM you might consider migrating to DQM, see my blog on [migrating from CQM to DQM](#).

The location of images is also different and you can refer to the [documentation for details](#). For the samples to work properly the images are now in `<install_location>/webcontent/bi/samples/images`. So during the migration planning make sure to account for all the images that have been used on the reports.

Another topic that me be encountered is customization. In the past this was referred to as branding but with Cognos Analytics the capabilities have been expanded. Samples are included as a reference to demonstrate the functionality of themes, extensions and views and are [documented here](#). Themes are used to reflect your corporate branding of the Cognos Analytics interface. Extensions can be used to add or remove items from the interface. Views can be used to modify the sign-in and default home for users. Additionally, control on which features are exposed to users can also be managed using role customization and is [documented here](#).



# Establishing a Baseline

## The Plan

The success of an upgrade can be measured based on the execution of a test plan. Building a test plan consists of 2 critical steps.

1. What reports need testing
2. What scenarios need testing

Determining what reports need to be tested should be based on an agreement with the user community and gaining their confidence that sufficient testing confirms that their reports will work in the new environment. Does that mean 100% of their reports need testing or is there a sample of reports that would suffice?

When you collect an inventory of reports you need to make a few decisions.

- Are there duplicates that can be eliminated?
- Are there reports that can be optimized or modernized? Perhaps these are opportunities for another project. It is recommended to keep the upgrade simple and leave this type of effort for a different project.
- Are all these reports still needed? Consider archiving reports that are not needed then if they are requested later you can retrieve them from the archive.
- Do all these reports still work?

Some discussion needs to occur on the contents of the reports in the “My Folder” as opposed to those in the “Public Folder”. An argument could be made that anything in the “My Folder” should remain the responsibility of the owner and anything that was placed in a shared folder could be considered for the overall test plan. In the end there are restrictions to accessing reports in “My Folder” which could edict their exclusion from the overall test plan.

Determining how the reports should be tested includes a few more decisions.

- Do the reports have multiple use cases that require different tests?
- Do you test against production data or is there a testing/staging database that could be used?
- Do you test on your production Cognos environment or should you stand up a sandbox environment for testing?
- Is the data frozen such that data refreshes or reports with relative period requirements skew the results? This is more relevant on when the target execution occurs and for the output comparison, more on that in a future blog.
- Do you need to test with different security roles?
- Do you have the parameter values for reports with required prompts?

In the end a good test plan will define what needs to be tested and how to test those reports.

### **The Baseline**

Testing the at the source is the beginning. Here source is defined as the current environment. After you have defined the plan of what reports need to be tested and what scenarios to test you establish a baseline. The baseline would include all the test cases to satisfy the end user test requirements and generated output for each of those test cases in the source environment. That output would be used during the validation phase. Basically, the validation phase would use the same test cases but in the target environment and then compare the two outputs. Here target is defined as the new environment/version that we are targeting the upgrade for. More on that in a future blog. The goal of the baseline is to make sure the reports work before migration occurs for each of the test cases defined.

### **Manual or Automated Testing**

How do you execute all those test cases? Depending on how many reports and test cases your plan includes it might be possible to complete the testing using a manual approach. Scalability and reuse for regression testing would be an obvious value in using automated testing tools. Regression testing could be considered extremely valuable particularly with the frequency of releases of Cognos Analytics. Consideration for the resources required to complete the testing does need to be taken into account. If a manual approach is used, how long will it take to execute the test plans? In contrast if you use an automated tool what is the resource load on the environment being used so that the production environment is not impacted, unless you planed on using a sandbox environment and a staging/test database. Additionally, most automated test tools will not test javascript as they would only validate the data or the rendered output.

When looking at the timeline for testing both the source and target you need to consider the impact of change requirements. In most environments even if you have a solid change management process it might be challenging to enforce a code freeze for the duration. Tracking changes such that retesting can occur is either a manual process or some testing tools also have the ability to track stale reports. It can be frustrating to test a source report that has changed but not migrated to the target environment or to compare the output of a report that has since changed.

### **Summary**

Knowing what and how to test before migration is important. Fix, delete or archive in the source environment before migration. Generate the output of those test cases for use in the validation phase.

# Content Migration

## The Contents

With confirmation of which reports to migrate and having executed the baseline we need to get that content into the target environment. Two options exist including Content Store database copy or using Export/Import capability from the administration panel. Notice I did not mention an over the top install, although that is a supported approach from an application perspective it is not recommended for a migration strategy. Obvious reasons include there would be no way to compare the before and after reports to confirm the upgrade was successful and losing the original source environment should anything go wrong. For large environments with many reports it might be easier to use a database copy process. Once the new environment is configured and pointing to this copy the table structures that support the content store would be automatically upgraded to conform to the new schemas. Using an export/import process could be accomplished either using the entire Content Store or selected subsets, possibly organized by subject area. When using the Export/Import process and you have a large Content Store you may run into issues during the export that might benefit from some tuning including the memory allocated to the process and there are some advanced parameters that can also help. Here is a tech doc that explains some of these [Link to tech doc](#). When using the export/import process a consideration is to make sure all the dependencies are also included. Items like packages, security roles and groups and data source configurations.

## Upgrading the Report Specifications

With each release of Cognos it is possible that the Report Specification schema changes. It is recommended to use the Content Maintenance Task to upgrade the Report Specification. This applies to both methods of migrating the Contents as discussed above. The logging provided from the execution of the Maintenance Task can then be leveraged to identify any errors that may need resolution. This process will not modify the “interactive flag” which would change the report from rendering with the classic viewer to the new interactive viewer. More on that in a future blog on Modernization. Note that if the specification is using an old scheme (created in a previous version of Cognos) and the upgrade is not done then each report specification may need to be upgraded each time when it is executed.

## Migration Pre-work

It is beneficial to disable all scheduled tasks such that they do not execute until you have confirmed the migration. If you disable them in the source environment before the migration, then you won't need to rush in the target environment to avoid unnecessary executions and high database utilization.

Some assets may be broken in your environment particularly if the environment has been in use for a long time. Things to look for are reports that do not have a valid package connection, report views that are not linked to a report and objects (especially schedules) that do not have valid members. Some of these can be cleaned with a Content Administration task or utilities that the Lab Services teams have access to. Some of these may also be identified during the import process.

After the target environment is setup make sure all the required data sources work properly before testing reports. Part of this is confirming the correct drivers have been installed and also that all the versions of the databases are supported which can be confirmed on the [supported environments website](#).

# Validation and Remediation

## Report Specifications

Part of the migration is to upgrade the Report Specifications. Hopefully that was completed as part of the migration phase. One step that should be done is the report spec validation. I have never (yet) seen a report spec validation fail as a result of the migration. In other words, if the report spec validation was successful in the source environment it will also be successful in the target environment. A recommendation then is to perform the report spec validation in the source environment and fix those that have failed there before the migration.

## Generating Output for Comparison

Using the test cases from the Baseline we can execute the same tests in the target environment. Some decisions need to be made depending on the type of report and what you intend to compare to determine a successful upgrade. If a visual rendering of the report is important the best comparison currently is the PDF rendered output. PDF comparisons using the automated tools can detect differences in layout and flag reports that may require further analysis to determine success or failure, the tolerance for difference needs to be part of the decision. If only the data is important then a data comparison can be performed. Lifecycle Manager will compare the data for each query in the report and MotioCI can be configured to compare the query data or the visual container data.

## Manual Comparison

If you are not able to use an automated testing tool for this there are some options to aid a manual comparison process. I have used some open source PDF comparison applications as well as command line scripts to compare CSV output for data comparisons. The manual process is not as scalable therefore it is recommended to look at the automated tools in the planning phase.

## Difference Comparison

If the migration is simple as defined in my planning blog then the success rate based on helping many customers with their migration is ~99%. There are however false negatives that you should be aware of when using the automated testing tools for comparison. The rate of false negatives averages 24% and can be categorized as *Pixel Shift* or *Sorting Difference* errors.

A *Pixel Shift* is when the data is consistent, but the rendering may result in an offset and is flagged by the automated comparison tools because they are using an overlay bit comparison process. These images are

**International Exchange**

examples of the overlay where the rendering is just offset in either a text item or a simple list.

PSD-Generation Date
11/14/2017
11/14/2017
11/14/2017
11/14/2017
11/14/2017
11/15/2017

Here is a slightly more complex shift that results in the data wrapping differently and ultimately may result with a different number of pages.

Interdept Unallowable Office Supp-Non-Book Ctr	Interdept Unallowable Office Supp-Non-Book Ctr
--	---

Upon farther analysis using a manual comparison you could flag these reports as successful or if the rendering itself is not critical you might rely on the results of a data comparison.

*Sorting differences* also occur and would be flagged when using the rendering comparison. This would occur on reports where the sorting is not specified to a unique order for the data. This is a result of the database optimizing the query path differently based on the SQL generated between the two versions of Cognos Query Service and in some cases returning the data back to Cognos is a different order. These images are an example, on the left is the source report and the right is the target report where you can conclude with manual comparison that this is a successful migration. Here too if the rendering is not critical you might rely on the results of a data comparison.

Subctr:D/C Subcontracts	7,978,828
AC 3	8,083,904
AC 3	8,060,472
AC 3	8,132,872
AC 3	8,107,367
AC 3	7,970,479
Subctr:D/C Subcontracts	7,888,802
AC 3	7,994,233
AC 3	8,038,271
AC 3	7,948,390
AC 3	8,016,099
Subctr:D/C Subcontracts	8,069,233
AC 3	7,924,581

AC 3	7,924,581
AC 3	7,948,390
AC 3	7,970,479
AC 3	7,994,233
AC 3	8,016,099
AC 3	8,038,271
AC 3	8,060,472
AC 3	8,083,904
AC 3	8,107,367
AC 3	8,132,872
Subctr:D/C Subcontracts	7,888,802
Subctr:D/C Subcontracts	7,978,828
Subctr:D/C Subcontracts	8,069,233

Data Comparison is particularly useful when the data is of concern and the rendering is insignificant. Unfortunately, many reports are still being used for ETL or just to generate a CSV output. Those instances are candidates for data comparison and you should forego the rendering comparison to simplify the process.

Here are the results of data comparison using Life Cycle Manager

SOURCE/TARGET	EXECUTION TIME	FORMAT	RUN STATUS	COMPARISON STATUS	DETAILS
Source	1.865	data	✔		
Target	2.636		✔	❗	<a href="#">Show details</a>

Here are the results of data comparison using MotioCI

```
Component Data Comparison Failure 0.390s [List4] outputs are identical (Component Data Output Diff).
[UserLogins] outputs are identical (Component Data Output Diff).
[ReportUsageQuery] outputs are different (starting on page 1) (Component Data Output Diff).
[List3] outputs are identical (Component Data Output Diff).
[List1] outputs are identical (Component Data Output Diff).
[PackageUsageQuery] outputs are identical (Component Data Output Diff).
[UserReportExecutions] outputs are identical (Component Data Output Diff).
[List2] outputs are different (starting on page 1) (Component Data Output Diff).
```

Other factors may impact the comparison. You need to determine if the data is dynamic or static and plan accordingly. Dynamic can be in two forms, either the database is updated or the report is using relative time periods. If the data is dynamic then you need to execute the source and target reports within a time window where the data is consistent.

Additionally, if the reports have “run date” or “run time” these too could fail a comparison. MotioCI can be configured to ignore those type of data elements.

**Results and Remediation**

Having executed the test cases in the Target environment you should have a good understanding of the success of your migration efforts. After reviewing the reports that are flagged as failed you should also have been able to categorize them as false negatives as described above or as needing remediation. Hopefully at this point the number of reports that actually require remediation is low.

Here are the test results using Life Cycle Manager



Here are the test results using MotioCI

**79 Test Cases failed.**  
**31 Test Cases have warnings.**  
**568 Test Cases have run successfully.**

## Go Live

### After the migration, what's next?

- If you had disabled the schedules they should be enabled for those reports that have been migrated.
- It would be beneficial to keep both environments running in parallel for a period of time. The old environment could be restricted and only used to investigate issues should any arise.
- If you are migrating in batches you should enable users in the new environment that align to those reports that have been migrated. Depending on your security concerns you could provide all your users access to the new environment. Remember the licensing has changed and there are new tools included in the default roles, you may need to revisit how the user access is defined.
- If your users are not familiar with Cognos Analytics then some education would be beneficial.

### Training

If your previous environment was C10 or earlier it would be important to provide an introduction to the new portal. Report Authoring has changed from Cognos 10 to Cognos Analytics 11 and again in Cognos Analytics 11.1 so your training should be specific to the release.

Reports that have been migrated should be accessible with some basic training, here are some links to get started.

Cognos Analytics –YouTube Channel

<https://www.youtube.com/user/Analyticszone/featured>

<https://www.youtube.com/user/IbmBaEducation/featured>

Library of How-To Videos for IBM #Cognos Analytics

<https://community.ibm.com/community/user/businessanalytics>

Demos and Hands on Labs

<https://www.ibm.com/demos/collection/IBM-Cognos-Analytics/>

### Modernization

After Go-Live with the existing content you should build a strategy on how to modernize your analytics. That sounds complicated but the implementation is actually simple. The new features of Cognos Analytics are making it easier to build modern solutions. The direction of Adhoc reporting is changing from what is now labeled Legacy Studios to various components of the Cognos Analytics environment including

Dashboards, Exploration and Stories. Dashboards provide an interactive and visual data discovery capability. Exploration is the AI assisted ad-hoc data discovery tool. The new licensing of Analytics Users has access to many of the tools.

The new licensing of Analytics Users has access to many of the new tools.

Reports that have been migrated will have the same behavior as they did in the previous version of Cognos where new reports will default to the interactive user interface with the pop-up menus. It is not recommended to bulk update all reports to use the new interactive features for several reasons. The most complex reason for not automatically updating a report to the new interactive viewer is the implementation of JavaScript. [Here](#) are some details on using JavaScript in Cognos Analytics.

A future Blog will cover modernization in more detail.

If you need help with your migration please contact [IBM Expert Labs](#) or [Cognos Analytics Advocates](#)