

Managing Actuate Report Design Source Files

By Chris Geiss
chris_geiss@yahoo.com
<http://www.chrisgeiss.com>

Managing Actuate Report Design Source Files

By Chris Geiss
Revised 5/1/02

Copyright ©2002, 2001 by Chris Geiss. All rights reserved. This document may be redistributed providing that the document is distributed unmodified and intact, with all copyright notices preserved. Information in this document is subject to change without notice.

This document is not affiliated with, nor endorsed by, Actuate Corporation. Actuate, e.Analysis, e.Report, e.Reporting, Live Report Document, Live Report Extension, ReportBlast, ReportCast, Report Encyclopedia, SmartSearch, Transporter, Virtual Report Distribution, and XML Reports are trademarks or registered trademarks of Actuate Corporation. Microsoft, Windows and Windows NT are trademarks or registered trademarks of Microsoft Corporation. All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

This document is being provided free of charge to my clients, and to the Actuate user and developer communities. This document is provided as is. No warranty or guarantee, either expressed or implied, is made about the suitability of this information to any particular application. Every reasonable attempt has been made to ensure that the information contained in this document is accurate. However, no guarantee is made that the information contained within this document is free from errors or omissions. Use this information at your own risk.

Table of Contents

About the Author.....	1
The Mailing List and How to Report Errors and Omissions.....	1
Do You Have Suggestions for a New Article?.....	1
Introduction.....	1
Recommended File Names	1
Folder Organization for Source Files.....	1
Example Folder Structure for an Organization.....	4
Important Points About Source File Organization	5
Using Relative File References	5
Folder Organization For Different Environments	5
Version Control.....	7
Manual Version Control System	7
Manual Version Control Process.....	9
Deployment Considerations.....	10
Version Control Software.....	11

About the Author

Chris Geiss is an independent consultant and software developer with over 16 years of software development and IT industry experience. He has been working with the Actuate reporting products for 4 years and specializes in consulting, development, training and mentoring services for the Actuate e.Reporting Suite. Chris can be contacted at chris_geiss@yahoo.com. See his web site at <http://www.chrisgeiss.com>.

The Mailing List and How to Report Errors and Omissions

If you find any errors or omissions in this document, or have any suggestions, please email them to chris_geiss@yahoo.com so I can update this document accordingly. Also, if you email your contact information to me, I will include you on the mailing list for future updates to this document.

Do You Have Suggestions for a New Article?

If there is a particular Actuate related topic that you would like more information on, please email me and let me know. If the topic seems of interest to the general Actuate developer and user communities, I may base a future article on it.

Introduction

This document discusses the various issues involved in the management of the source files that comprise your Actuate report designs, as well as recommendations on how to organize your design files and handle source code version control for your report designs.

Recommended File Names

The following guidelines are recommended when naming your report design source files:

- When naming .rod files, use names that would be meaningful to a user, since the users will use the file names they see via ReportCast (or other client side interfaces) to decide which report they wish to run; for example, instead of using the name "SalesRpt001.rod", name it "Sales Performance Report.rod"
- Use spaces instead of underscores in the .rod names so that the names will appear better when viewed via ReportCast and a web browser; for example use the name "Account Summary.rod" instead of "Account_Summary.rod." Report development is done on Windows based systems, which allow spaces in file names. The Actuate Report Encyclopedia allows spaces in report object names whether it is running in a Windows or UNIX environment.

Folder Organization for Source Files

This section outlines the recommended organization for the folders that are used to hold the source files that comprise your Actuate report designs. The specific folder names are not important, but the folder hierarchy is recommended for the purposes outlined below. Indentation represents the folder levels.

Folder Name	Purpose/Description
\Actuate Reports	Main (top level) folder containing the source files for all report designs in your organization or department. This folder can be placed in the root of your file system, but does not have to be.
\Common	Generic base reports, libraries, and basic source files that can be used in any report design that is created for your organization or department.
Base_Report.rod PageLayouts.rol Global_Functions.bas	Example file names.
\Images	Images and graphics that are shared by all report designs.
\App_1	The source files for all the report designs that belong to "Application 1." Application boundaries might be drawn according to the departments the reports are for, the ERP applications they are to be integrated with, etc.
\Common	Base reports, libraries, and basic source files that are specific to the report designs associated with "Application 1."
App_1_Base_Report.rod App_1_Components.rol App_1_Functions.bas	Example file names.
\Images	Images and graphics that are shared by all "Application 1" report designs.
\App_1_Report_1	The design files specific to the first report design for "Application 1." This will include the .rod, .bas, .rox and .roi file for the design as well as any Basic source files or libraries that are used only by this report design.
App_1_Report_1.rod App_1_Report_1.bas App_1_Report_1.rox App_1_Report_1.roi App_1_Report_1_Lib.rol App_1_Report_1_Code.bas	Files associated with the first report design for this application.
\App_1_Report_2	The design files specific to the second report design for "Application 1."
App_1_Report_2.rod App_1_Report_2.bas App_1_Report_2.rox App_1_Report_2.roi App_1_Report_2_Lib.rol App_1_Report_2_Code.bas	Files associated with the second report design for this application.
\App_2	The source files for all the report designs that belong to "Application 2."
\Common	Base reports, libraries, and Basic source files that are specific to the reports associated with "Application 2."

Folder Name	Purpose/Description
App_2_Base_Report.rod App_2_Components.rol App_2_Functions.bas	Example file names.
\Images	Images and graphics that are shared by all “Application 2” report designs.
\App_2_Report_1	The design files specific to the first report design for “Application 2.”
App_2_Report_1.rod App_2_Report_1.bas App_2_Report_1.rox App_2_Report_1.rol App_2_Report_1_Lib.rol App_2_Report_1_Code.bas	Files associated with the first report design for this application.
\App_2_Report_2	The design files specific to the second report design for “Application 2.”
App_2_Report_2.rod App_2_Report_2.bas App_2_Report_2.rox App_2_Report_2.rol App_2_Report_2_Lib.rol App_2_Report_2_Code.bas	Files associated with the second report design for this application.
\App_3	The pattern outlined above repeats as needed for each application.
\Common	
\Images	
\App_3_Report_1	
\App_3_Report_2	
\App_3_Report_3	
\App_3_Report_4	

Example Folder Structure for an Organization

The folder outline below shows the actual folder names and sample report design file names that might be used by an organization based on the general folder structure presented above. In this case, applications have been partitioned according to the departments within the organization.

```
\Actuate Reports
  \Common
    Internal_Use_Base_Report.rod
    External_Use_Base_Report.rod
    Portrait_Mode_Page_Layouts.rol
    Landscape_Mode_Page_Layouts.rol
    Math_Functions.bas
    Text_Functions.bas
    etc.
  \Images
    Corp_Logo.jpg
\Accounting Reports
  \Common
    Accounting_Page_Layouts.rol
    Accounting_Functions.bas
  \Images
    Acct_Rpt_Banner.jpg
  \Account Detail
    Account Detail.rod
    Account Detail.bas
    Account Detail.rox
    Account Detail.roi
  \Account Summary
    Account Summary.rod
    Account Summary.bas
    Account Summary.rox
    Account Summary.roi
    Account Summary_Lib.rol
    Account Summary_Code.bas
\Sales Reports
  \Common
    Sales_Page_Layouts.rol
    Sales_Functions.bas
  \Images
    Sales_Rpt_Banner.jpg
  \Sales Rep Performance
    Sales Rep Performance.rod
    Sales Rep Performance.bas
    Sales Rep Performance.rox
    Sales Rep Performance.roi
    Sales Rep Performance_Lib.rol
    Sales Rep Performance_Code.bas
    Sales Rep Performance_Banner.jpg
  \YTD Sales
    YTD Sales.rod
    YTD Sales.bas
    YTD Sales.rox
    YTD Sales.roi
```

Important Points About Source File Organization

The following are some important points and ideas about the folder structures outlined above.

- It is not necessary to have all report designs under a single “\Actuate Reports” folder, but this may be desirable in some cases. For example, if there is only one set of developers working on related applications, then it might make sense to have only one “\Actuate Reports” folder. However, if there are multiple developer groups, perhaps in different geographic locations, working on distinctly separate applications, then it might be appropriate for each development group to have its own “\Actuate Reports” folder structure.
- Each report design is kept in a separate folder. This makes it easy to keep the source files for a given report design together and therefore makes it easy to move them from one place to another, or to back up the source files for a single report design (for example, using a zipping tool such as WinZip).
- Base reports, libraries and Basic source files are kept in appropriate folders in the folder hierarchy. They are either kept in a specific report design folder if the files only apply to one report design, or in a “Common” folder at the appropriate level of the folder hierarchy according to the group of report designs for which they are intended (application level, global level, etc.)
- **All report designs and “Common” components must use relative file paths when linking to other files.** This makes it possible to relocate the entire folder structure from one location to another (such as from the root folder “\Actuate Reports - DEV” to the root folder “\Actuate Reports - PROD”) without breaking any file references. It will also make it easier to move a specific report design from one folder structure to another without having to resolve all file references, although some may have to be changed. **It is very important that you consider which file references need to be updated when you move a report design from one location to another.** However, so long as the relationship of the folders in the hierarchy does not change, file references should not have to be updated.

Using Relative File References

As mentioned above, whenever files are linked to a report design (using the Library Organizer, or linking to image files, etc.), only relative file paths should be used. For example, if the “Account Detail.rod” report design mentioned in the example above needed to link to the “\Actuate Reports\Common\Images\Corp_Logo.jpg” file, it would reference the file as “..\..\Common\Images\Corp_Logo.jpg”. The double dot notation (..) means to go up one level from the current folder. Thus, the path just given says to go up two levels from the folder containing “Account Detail.rod”, which would bring us to the “Accounting Reports” folder and then to the “\Actuate Reports” folder. From there, the “Common\Images\Corp_Logo.jpg” file would be found. Since there is no specific drive letter referenced, and the names of the top two folder levels are not mentioned, it becomes much easier to move the “Account Detail.rod” design files to another folder location, providing that the folder that contains them maintains the same hierarchical relationship to the “Common\Images\Corp_Logo.jpg” file. As another example, if the “YTD Sales.rod” design file needed to be linked to “\Actuate Reports\Sales Reports\Common\Sales_Page_Layouts.rol”, it would reference the file as “..\Common\Sales_Page_Layouts.rol.”

Folder Organization For Different Environments

If your organization uses a formal development life cycle, where report designs are phased from one stage to another, the folder structure proposed above can still be used. For example, if the report designs in your organization are transitioned from the development group, to the Quality Assurance (QA) group to the production environment, one of the following two folder schemes could be used to manage the files:

FOLDER SCHEME 1: Separate file server for each environment

```
\\Dev Server
  \Actuate Reports
    \Common
    \App_1
      \Common
      \App_1_Report_1
    \App_2
      \Common
      \App_2_Report_1
\\QA Server
  \Actuate Reports
    \Common
    \App_1
      \Common
      \App_1_Report_1
    \App_2
      \Common
      \App_2_Report_1
\\Prod Server
  \Actuate Reports
    \Common
    \App_1
      \Common
      \App_1_Report_1
    \App_2
      \Common
      \App_2_Report_1
```

FOLDER SCHEME 2: All files on the same file server

```
\Actuate Reports - DEV
  \Common
  \App_1
    \Common
    \App_1_Report_1
  \App_2
    \Common
    \App_2_Report_1
\Actuate Reports - QA
  \Common
  \App_1
    \Common
    \App_1_Report_1
  \App_2
    \Common
    \App_2_Report_1
\Actuate Reports - PROD
  \Common
  \App_1
    \Common
    \App_1_Report_1
  \App_2
    \Common
    \App_2_Report_1
```

Note, with the above folder structures, report designs could be easily moved or copied from one environment to the other, and recompiled without fixing file references, as long as the same folder structure is maintained in each location and files are linked to the report designs using relative file paths.

Version Control

Version control is a very important tool for managing the changes that inevitably occur to source code files. There are several objectives for version control:

- Make sure you know what source files were used to build the executables (rox's) currently deployed to production.
- Track the changes that occurred over time to the source files for any given executable; this makes it possible to see how the source files evolved over time as well as to refer back to earlier source code if a change made in a later version has introduced a new bug.
- Having the ability to "go back in time" and rebuild an executable as it existed in an earlier point in time; this makes it possible to roll back to an earlier version of an executable if the current one is not functioning properly.
- Implementing a "semaphore" system which makes sure that only one developer can be working on changes to a given source file at any one time. This also makes it possible to track who is working on a file if another developer needs to access it.

Version control can either be handled manually, or using a version control software tool. **Whenever possible, it is highly recommended that a version control software tool be used to avoid the problems (such as user error) that are inherent in any manual, human-based system.**

If a version control software tool is used, then the folder structure used in the version control tool can simply mirror the folder structure that has already been outlined. Doing so will make it very easy to check source files in and out of the version control system.

Manual Version Control System

If a manual version control system is going to be used, then the folder structure outlined above has to be modified to accommodate the ability to track different versions of a given report design or library component.

The following objectives were considered as the folder structure below was created:

- It should be easy to identify the source files for each version of a report design
- It should be possible to easily recompile an earlier version of any report design executable
- It should be easy to transition a report design through the life cycle process (for example from development to QA to production) by moving or copying the report design files from one folder location to another (for example from the development server to the QA server)
- Common report components should also be under version control

Folder Name	Purpose/Description
\Actuate Reports	Main folder containing the source files for all versions of the report designs in your organization.
\Common	The most recent version of the generic base reports, libraries, and Basic source files that can be used in any report design that is created for your organization. This folder will contain a copy of the files in the most

Folder Name	Purpose/Description
	recently dated folder below; in this case (“Common 2001-10-16”).
\Common 2001-05-31	All of the files that were released in \Common on 5/31/01.
\Common 2001-10-16	All of the files that were released in \Common on 10/16/01.
\App_1	The source files for all versions of the report designs that belong to “Application 1.”
\Common	The most recent version of the base reports, libraries, and basic source files that can be used for reports associated with “Application 1.” This folder will contain a copy of the files in the most recently dated folder below (“Common 2001-10-25”).
\Common 2001-10-01	The source files that were released in \Common on 10/1/01.
\Common 2001-10-25	The source files that were released in \Common on 10/25/01.
\App_1_Report_1 2001-10-05	The design files specific to the first report design for “Application 1” as it was released on 10/5/01.
\App_1_Report_1 2001-11-10	The design files specific to the first report design for “Application 1” as it was released on 11/10/01.
\App_1_Report_2 2001-10-05	The design files specific to the second report design for “Application 1” as it was released on 10/5/01.
\App_1_Report_2 2001-11-10	The design files specific to the second report design for “Application 1” as it was released on 11/10/01.
\App_2	The source files for all versions of the report designs that belong to “Application 2.”
\Common	The most recent version of the base reports, libraries, and basic source files that can be used for reports associated with “Application 2.” This folder will contain a copy of the files in the most recently dated folder below (“Common 2001-09-16”).
\Common 2001-09-16	The source files that were released in \Common on 09/16/01.
\App_2_Report_1 2001-10-01	The design files specific to the first report design for “Application 2” as it was released on 10/1/01.
\App_2_Report_1 2001-11-18	The design files specific to the first report design for “Application 2” as it was released on 11/18/01.
\App_2_Report_2 2001-10-05	The design files specific to the second report design for “Application 2” as it was released on 10/5/01.
\App_2_Report_2 2001-11-21	The design files specific to the second report design for “Application 2” as it was released on 11/21/01.

The following important points relate to the folder structure outlined above:

- **Any file that is linked to a report design MUST be linked with relative file paths, and not absolute file paths, or this manual version control system WILL NOT work.**
- Each new version of a design, whether a report design, or a folder of common components, is kept in a separate folder which specifies the date on which the design was released
- There will be a \Common folder at each level, which will **not** have a date in the folder name, and which will have a copy of the files in the most recently dated \Common folder at that level. This is done because the report designs will have references to the “Common” folder. If there was no “Common” folder without a date in the folder name, then these report designs would not compile until all of the file path references were updated.
- It is assumed that every time a new version of the files in a “Common” folder is released, that the new version will be designed to be backwards compatible with all earlier versions of the components. Thus a report design that was designed to use an earlier version of files in a “Common” folder should work fine when compiled with the latest version of the files in that “Common” folder.
- It is assumed that by default, whenever any version of a report design is compiled, that it will be linked to the most recently released versions of any “Common” files that it uses. This is a logical assumption since “Common” files are meant to be shared by report designs, and when a “Common” folder is updated, it is generally desired that the new changes be incorporated in all report designs that use those files.
- It is also assumed that all of the components in a “Common” folder are treated as a single set of files. Thus, even if only one file in a “Common” folder is modified, then all of the files in that folder will be released in a single group as a new version of the components. The new files will be released in a “Common” folder with the appropriate date in the file name and also copied to the “Common” folder that contains the latest version of the files (and does not have a date in the folder name).
- Every time a new version of a “Common” folder is released, a new folder should be created with the date of release, and the files should be copied to replace the current contents of the “Common” folder at that level which does not have a date on it.
- Assuming that relative file paths are properly used to link source files to report designs, the folder structure above will make it possible to rebuild any previous version of a report design for which the source files have been versioned and preserved. Note, that unless you take specific actions, the report design will be compiled with the most recent versions of the “Common” components that it uses and not necessarily the versions that were available when that version of the report design was first created or released.
- You can locate the specific versions of “Common” files that were used by a given version of a report design by examining the dates on the specified folders. For example, you can tell by examining dates, that the report design in the “App_1_Report_2 2001-10-05” folder would have used the common files in “\Actuate Reports\App_1\Common 2001-10-01” and “\Actuate Reports\Common 2001-05-31.”
- If a given environment does not need to track multiple versions of the report designs, then it is not necessary to use dates in the folder names. For example, the development environment might only contain one version of each report design -- namely, the version currently being worked on. Other environments, such as the production environment, might contain all versions of the report designs.

Manual Version Control Process

If a manual system is being used to version control your report design source files, then the following version control procedures are recommended:

- When a set of files, such as a report design, or a folder of common components, is moved into version control, the new set of files should be placed in a folder which is named according to their purpose and the date on which they are being released. For example, if common components

which apply to all reports are first released on 6/10/01, then they would be placed in the folder “\Actuate Reports\Common 2001-06-10”. If the components were subsequently modified and released again on 8/15/01, they would be placed in the folder “\Actuate Reports\Common 2001-08-15.”

- Whenever a new version of common components is released, they should be copied to replace the contents of the existing “Common” folder (which does not have a date in the folder name) at the appropriate level. This is done so that all report designs will automatically link to the new version of the components just released.
- In order to avoid problems with the accidental deletion of previous report versions, accidentally overwriting a previous version with a newer version, etc., it is highly recommended that only one or two people be designated as the “software librarians” and that they act as gatekeepers for anyone wanting to access or update any source code that is under version control.
- Only the software librarians should have write access to the version control file archives. Developers can have read-only access to these archives, although this may make it difficult to prevent two developers from working on the same report design at the same time.
- This process assumes that there are at least two primary source file repositories. A “development” area in which files currently under development are kept and are accessible to the developers. And a “master” or “production” area in which all versions of the report designs are kept and are only accessible to the software librarians.
- When a developer needs to update a report design, they should request the latest version of the source files from the librarian. The librarian should move a copy of the requested report design files from the master area to the development area, and then record that the developer currently has access to them. This will prevent the librarian from giving the same set of source files to another developer, until the first developer is done working on them.
- The developer will make the required modifications in the development area.
- When the developer is done with the changes, they will notify the librarian. The librarian will then move the new report design version from the developer area to the master area and update the log to indicate that the developer is no longer working on that report design. This makes the source files available to other developers.
- The same procedure that applies to report designs also applies to files contained in a “Common” folder.
- Whenever changes are made to the contents of a “Common” folder (meaning a new version has been released), the librarian will notify all developers that a new version is available and will copy the new version from the master area to the development area. The developers will then have to verify that their report designs still work with the newly released version of the “Common” files.

Deployment Considerations

Here are some important points to keep in mind when deploying report executables:

- Each report design is kept in a folder that has the same name as the .rod file. This makes it easy to find a given report design by scanning through the names of the folders. When a report design is compiled, the resulting executable file (.rox) will have the same base name.
- When a report is deployed on the Report Server, the .rox name should not be changed, so that it is easy to link a .rox on the Report Server with the report design source files that were used to create it. If you change the names of these .rox files, you will create major problems for people troubleshooting problems unless you are very meticulous about maintaining documentation that maps the names of .rox files on the Report Server with the corresponding design files that created them. Note, minor changes to .rox names, such as changing underscore characters to spaces to make the report names more user friendly, should not cause major problems.
- It is assumed that an .rox on a Report Server corresponds to the most recent version of the source files available for that report design in the production repository. This can always be verified by doing a binary file compare between the .rox file on the Actuate server and the one in the production repository.

- It is very important when you create a new report design to give the .rod file a name that will be suitable when the compiled .rox file is placed on the Report Server. For example, use the name “YTD Sales Report” instead of “Report010.”

Version Control Software

As mentioned earlier, it is strongly recommended that you use a version control software tool to manage version control for your report design source files -- especially if you are working on applications with many report designs. Affordable source control packages are available, as well as open source (free) version control tools. I have used Microsoft Visual SourceSafe very successfully with my Actuate report designs, but any similarly featured version control tool should suit your purposes just fine.