

Preparing to Load Your Application

Roundtable TSMS

Roundtable Development Team
Updated June 2012



Table of Contents

1	Overview	1
2	Analysis	2
2.1	Physical Application Components.....	2
2.2	Physical Component Relationships.....	2
2.3	Logical Grouping	3
3	Defining the Physical Structure.....	4
3.1	Workspace Module Definitions	4
3.2	Subtypes.....	5
4	Defining the Logical Structure.....	7
5	Loading the Application	8

1 Overview

Before loading your application into Roundtable Total Software Management System (Roundtable TSMS), some basic analysis is required. This collective process is a key step in implementing a software configuration management (SCM) solution for your development life-cycle as it will help to ensure an accurate and timely implementation while giving you the necessary data to perform configuration identification.

Accordingly, loading your application should consist of the following steps:

1. Analysis.
2. Defining the physical structure.
3. Defining the logical structure.
4. Loading the application.

This document offers guidelines on how to complete the critical but often overlooked steps in this process: (1) how to analyze your application directory, (2) how to create the corresponding physical structures in Roundtable TSMS, and (3) how to create the logical structures in Roundtable TSMS. It uses common examples to illustrate each case.

NOTE: This document is not intended to replace the Roundtable TSMS User Guide. It is intended to provide supplemental information to help make the process of loading the application easier. The examples do not cover every possible solution.

2 Analysis

Prior to loading, an analysis should be done on the baseline application directory that you wish to load into Roundtable TSMS. This analysis will assist you in identifying the key structural components of your application, the resources contained within, and the relationships between them. Depending on your familiarity of the physical structure of the application you wish to load, the analysis portion could be the most time-consuming part of the implementation process. It is during this phase that you will be performing the configuration identification step of the software configuration management process.

The purpose of the initial analysis is to help you answer the following questions:

- What are the physical components that make up my application?
- How do the physical components relate to one another?
- What are the logical groupings for my application?

This section can help you answer these questions, putting you well on your way to loading your application in the Roundtable TSMS managed environment.

2.1 Physical Application Components

At the most basic level, the physical components are the application directories and resources in those directories. However, your preparation should go a little deeper than that. Be prepared to answer the following questions:

- What types of resources are in each directory?
- What are the properties for each of those resources?
- Do I even need to load all the resources contained in each directory?

Having the answers to these questions will help you to identify the relationships between the physical resources and any special considerations that should be taken into account for managing those resources.

2.2 Physical Component Relationships

The relationships that may exist between individual resources should also be taken into consideration. Since Roundtable TSMS has the ability to manage groups of physical files as single entities, you should take the time to identify any physical resources that you may want to be managed as a single object. For example, an ADM2 SmartDataObject has three physical components (.w, _cl.w, .i) but Roundtable TSMS can ease the management of those components by treating them as a single entity. The same holds true for ABL GUI for .Net forms that have two parts (.cls, .resx).

2.3 Logical Grouping

Once you have identified all the physical components, you should decide how the resources logically fit together. For example, you may have many resources contained in a single directory but are they related logically? Do those resources act on completely different functional areas of your application? Going further along that train of thought, is your application really a single logical product, or is it multiple smaller products all residing in the same directory structure?

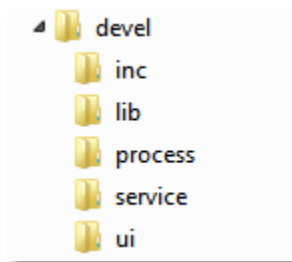
Being able to logically identify your application components is another step in the configuration identification process of SCM as these logical components will be gathered together to establish views of your application, which we call Workspaces. When defining your application life-cycle flow between those Workspaces later, having identified these logical components will greatly assist you in further establishing configuration control when defining which components participate in the promotion of changes.

3 Defining the Physical Structure

3.1 Workspace Module Definitions

Roundtable TSMS needs to know which physical directories make up your application so you will need to define them. Roundtable TSMS refers to these directories as Workspace Module Definitions and are defined using the Workspace Definitions Module maintenance interface. Workspace Module Definitions can also have attributes such as a unique name, description, and an r-code directory. Having the physical structures formally defined allows for greater control over where application resources are placed and also prevents rogue resources from being committed to the repository.

For example, let's consider the following simple application structure where 'devel' is the baseline development directory.



For each directory that contains physical resources, Roundtable TSMS requires that you create a Workspace Module Definition for it. If physical files exist in the five subdirectories, then a definition should be created for each directory.

Module	Description	Directory
Include	Include Files	inc
Library	Persistent Procedure Libraries	lib
Process	Business Processes	process
Service	Service Layer Procedures	service
User Interface	User Interface Components	ui

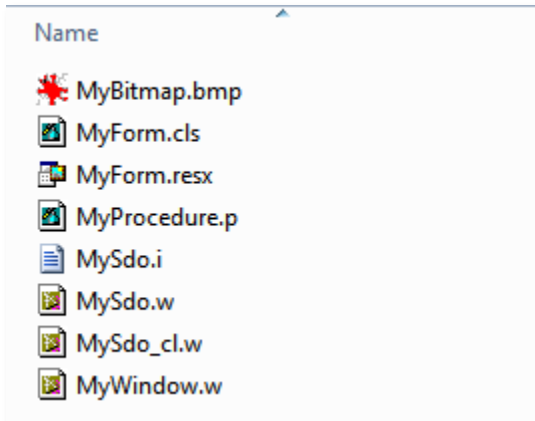
You can be as descriptive as you would like in your module naming convention. Optionally, you can specify an r-code directory if you would like any r-code created within the module to be placed into a different directory from the source directory.

3.2 Subtypes

Roundtable TSMS also needs to know what types of file you will be committing to the repository. Identifying physical file types helps you to further implement configuration identification and configuration control by pre-determining the types of resources that you wish to allow to be committed to your repository. This physical identification is done by defining code Subtypes.

In simple terms, code Subtypes allow you to give user-friendly names to file extensions while also allowing you to specify default properties (e.g., binary, compiles), identify compilable parts, and group physical files together for easier management and promotion.

Consider the following examples.



After a brief analysis, you can reach the following conclusions:

- A multi-part Subtype will be required for MyForm.* extensions (.cls, .resx).
- A multi-part Subtype will be required for MySdo.* extensions (.w, _cl.w, .i).
- A Subtype definition will be required for the .BMP extension.
- A Subtype definition will be required for the .P extension.
- A Subtype definition will be required for the .W extension.

Further analysis reveals two more conclusions:

- The 'binary' attribute should be set for the .BMP Subtype.
- The 'compiles' attribute should be set for the 2nd part of the SDO Subtype.

Based on these conclusions, the following Subtypes would be defined as follows:

Subtype	Bitmap
Parts	1
Extension	bmp
Binary	true

Subtype	Form
Parts	2
Extension 1	cls
Extension 2	resx
Compiles	true

Subtype	Procedure
Parts	1
Extension	p
Compiles	true

Subtype	Window
Parts	1
Extension	w
Compiles	true

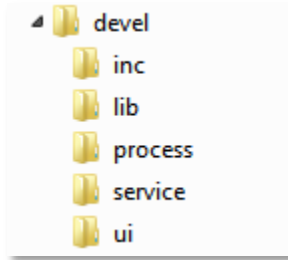
Subtype	SmartDataObject
Parts	3
Extension 1	w
Extension 2	w
Suffix 2	_cl
Compiles 2	true
Extension 3	i

NOTE: See the Code Subtypes maintenance interface for further attributes that can be set.

4 Defining the Logical Structure

Once you have identified all the physical aspects of your application, it is now time to define the logical layer that will be used within Roundtable TSMS. Roundtable TSMS allows you to identify functional areas of your application by breaking it down into Products and Product Modules. It's this logical grouping that Roundtable TSMS will use to help uniquely identify the resources that you have committed to the repository.

Let's go back to our physical directory example.



Using the Product Maintenance interface, you will need to define a Product and child Product Modules that logically organize your application. At a minimum, you will need to define a Product Module for each physical directory that contains resources that you wish to load into Roundtable. Each Product Module will be assigned a Workspace Module definition that you created earlier.

NOTE: Some directories may contain physical files that belong to different functional areas of your application. In those cases you may define multiple Product Modules that share a Workspace Module definition.

With prior example, the following Product and Product Modules would be defined as follows:

Product	Description
SportsPro	The Sample Sports Pro Product

Product Module	Description	Module Definition
sports-include	SportsPro Include Files	Include
sports-library	SportsPro Function Libraries	Library
sports-ar-process	SportsPro AR Processes	Process
sports-gl-process	SportsPro GL Processes	Process
sports-ap-process	SportsPro AP Processes	Process
sports-service	SportsPro Service Interface	Service
sports-ar-ui	SportsPro AR User Interface	User Interface
sports-gl-ui	SportsPro GL User Interface	User Interface
sports-ap-ui	SportsPro AP User Interface	User Interface

NOTE: To emphasize the possibilities, multiple Product Modules that share the same Workspace Module definition are indicated within the brackets above.

5 Loading the Application

Once the application structure has been defined, you can load the application resources into Roundtable TSMS and commit them to the repository.

Refer to the following sections of the Roundtable TSMS User Guide to complete the process: Adding a Workspace, Adding a Workspace Source, and Module Load.

While the configuration identification steps that we have just discussed can sometimes be the lengthiest part of your SCM implementation process, it can also be the most rewarding. Having a clear logical vision of your application and its functional areas helps you to improve your development processes by clarifying application boundaries and by giving meaning to application resources. Instead of just seeing the baseline as a collection of files in a directory, formally defining your product structure helps your team see the application as a collection of components, working together for the greater good.