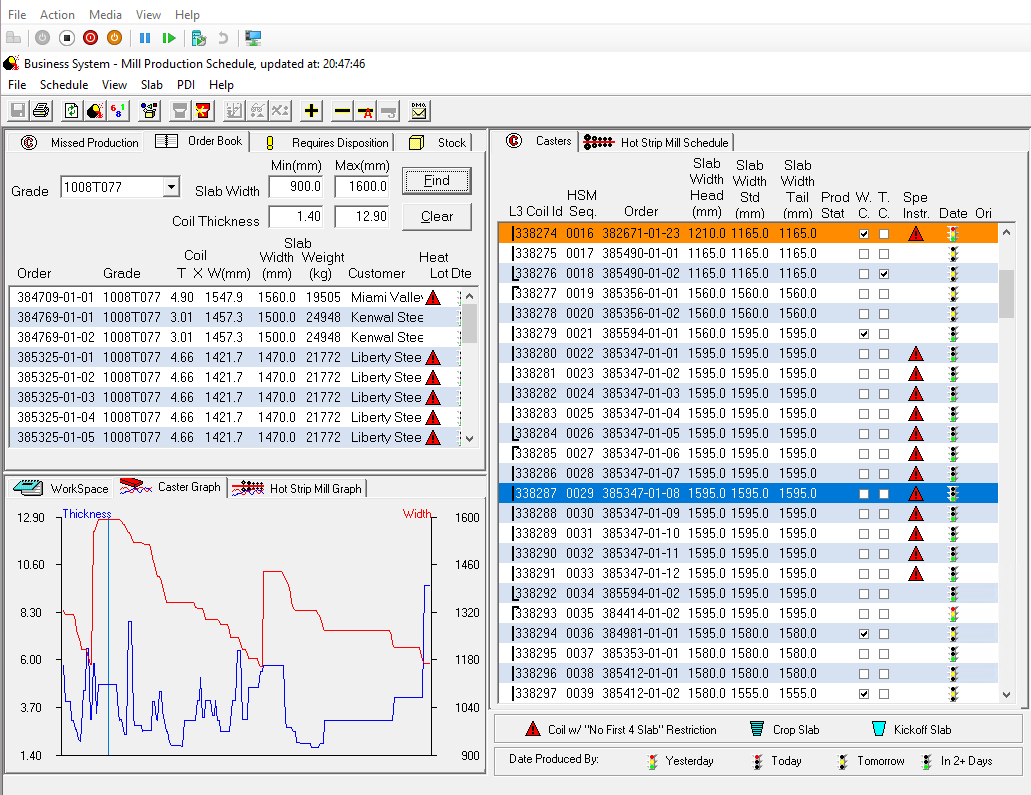
MPS and SR Documentation

Mill Production Scheduling and Slab Re-Application



Created by Dan Houck (AS&E, Ferret Software)

For North Star BlueScope Steel

Version 17.8.12

Table of  
Contents

Overview 1

Components and Reference 1

Quick Start Guide 2

Recommendations 3

First Step: Review the Overview ‘Tour’ videos 3

Second Step: Run the LogAnalyzer 4

DocPack Installation and Setup 5

Requirements 5

Folder Structure of the DocPack 6

Appendix – Instrumenting the C++ Programs 7

Appendix – Overview of the C++ Programs 9

MPS 10

RS 10

Support Code 10

Check\_Res 11

PPS\_LIB - LIB 11

LibSpc - LIB 12

SPCRES – DLL 12

Appendix – Getting the MPS and SR to Build 13

# Overview

This is a multi-media documentation package (which is referred to as a ‘DocPack’, or specifically, the MpsSr-DocPack. The DocPack strives to document the MPS and SR systems in a way that the developers of the replacement systems can more easily understand and explore the usage of the current system. It also provides a way to easily locate the detailed logic and code behind the usage of the MPS and SR systems.

The package comprises text documents, running systems, video/audio content, and an interactive LogTrace analyzer. This document is the Guide that will explain how to setup and use all components of this system.

## Components and Reference

Independent of this effort is the documentation of the Use Cases for these Systems. These are defined in the Functional Specification – SAP MII Phase 1.

This DocPack attempts to conform to the terminology and organization set forth in that document. Specifically, this DocPack attempts to connect the “Use Cases” described in the Functional Document to the actual usage of the MPS and SR systems, which allows a user of this DocPack to quickly examine the code/logic that drives the existing system.

## 

## Quick Start Guide

Because the users of this system are likely behind corporate firewalls that do not permit access to video distribution systems (such as YouTube), all content of this DocPack is available in a single file structure.

The DocPack can be used in the following ways:

1. Get an Overview of the systems by viewing the annotated video of operators using and explaining the system.
2. Run the LogAnalyzer against the provided TraceLog files for different Use Cases to see the logic flow and examine the code in detail.
3. Run the MPS and SR run-time systems and generate your own TraceLog Files to look at specific situations.
4. Run VS2003 and alter the TraceLog instrumentation to create new existing logging.

Item 3 requires that a System Administrator provide a login account and a way to access the Sybase database. The run-time system depends upon INI files that instruction the system how to connect to the SQL Database (which is Sybase).

The MPS system can be run by clicking on the Mps001.exe Icon under:

Source .MFC 2003 > Mps001.root > Mps001 > Debug.

The SR system can be run by clicking on the Sr001.exe icon under:

Source .MFC 2003 > Sr001.root > Sr001 > Debug.

Item 4 requires an operational version of Visual Studio 2003 and a good knowledge of the language/platform in which MPS and SR were written (C++/MFC).

## Recommendations

### First Step: Review the Overview ‘Tour’ videos

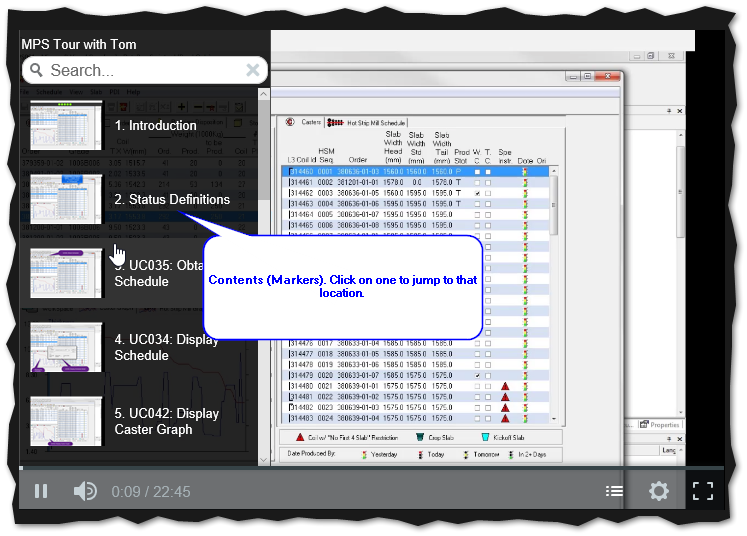
Start by going to the root of the DocPack (Mps-Sr DocPack), and then navigating to:

Content > Videos

There you will find two overview videos:

1. MPS Tour with Tom
2. SR Tour with Andrew

You can double-click on the MP4 or HTML files to begin the tour. If you use the HTML, a Table-Of-Contents is provided that allows the user to jump to marked sections of the video. Note the table of contents display/hides when the mouse hovers over the image.



## Second Step: Run the LogAnalyzer

A 6 minutes video tutorial for running the LogAnalyzer can be found in the MpsSr-DocPack under Content > Videos.

MPS-SR-LogAnalyzer Tutorial

This brief tutorial is an explanation of how the LogAnalzer can be used.

Under the DocPack’s TraceLogs folder are example TraceLogs (for example trace-UC033,034,035.log), but the user of the system will most likely wish to generate their own.

# DocPack Installation and Setup

The DocPack is simply a folder organization of the documentation and all of the support programs.

This section describes how to get going. It assumes that you have access to the MpsSr-DockPack.zip file.

## Requirements

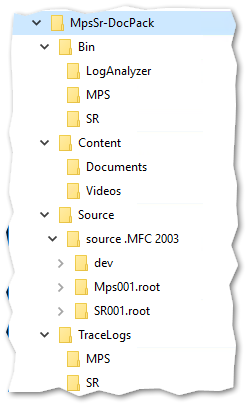
The DocPack must be used on a computer running Windows 8.1 or better. The free product Notepad++ should be installed. A player for video HTML/MP4 files should be present, as well as a PDF viewer.

Notepad++ is available for free download here:

<https://notepad-plus-plus.org>

## Folder Structure of the DocPack

The folder organization is as follows:



The Bin folder holds subfolders that can be used to run the three executables: MPS, SR, and the LogAnalyzer.

The Content subfolder holds documentation media, which includes PDF for print, PNG for pictures, and MP4 (launched by HTML) for videos.

The source holds each of the source files are displayed by LogAnalyzer with the help of the TraceLogs (which are under the Tracelogs folder).

# Appendix – Instrumenting the C++ Programs

Instrumenting the C++ programs involved adding two files, which reside under the folder dev \ cmTrace \ cm

1. Trace.cpp
2. Trace.h

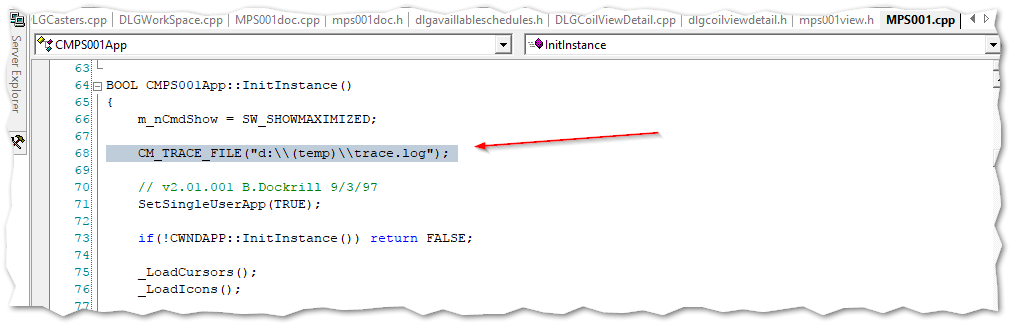
In each project, this involved adding the above folder’s location to the include path for the compiler.

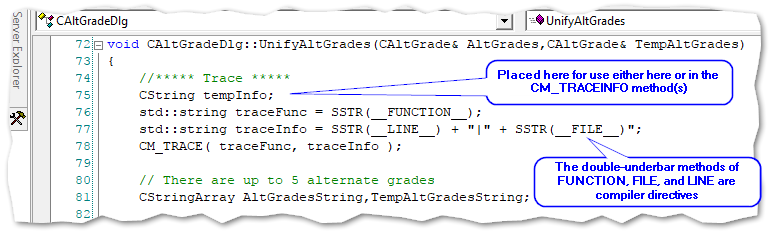
In each C++ file used, this involved adding the include “Trace.h” statement.

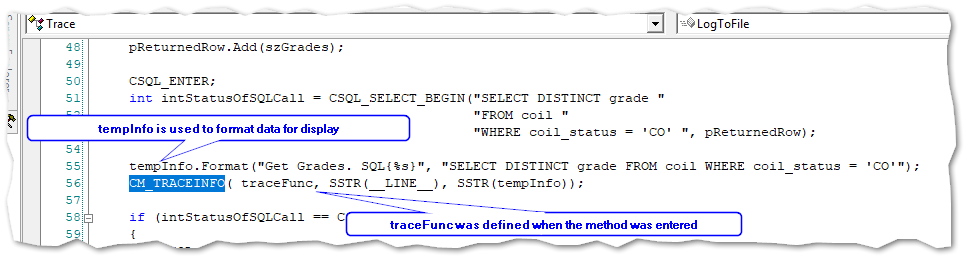
There are three methods used:

1. CM\_TRACE\_FILE is called only once – and placed at the beginning of logic – and sets the path of the trace log file.
2. CM\_\_TRACE is placed at the beginning of any method that needs to be traced.
3. CM\_TRACEINFO is placed within the method where additional information is required.

Examples:







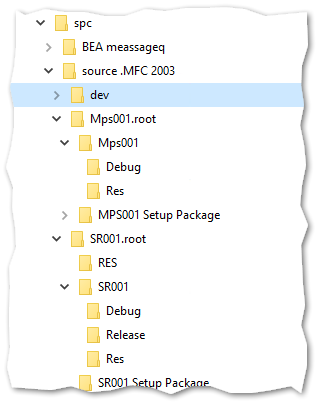
# Appendix – Overview of the C++ Programs

Both the MRS and SR systems are C++ programs built with VS2003 and using the Microsoft MFC (Microsoft Foundation Classes).

The MFC is a wrapper around the Windows API. It provides a C++ program with a simpler way to access key features of the Windows Operating System, including such things timers, file access, but most importantly the UI. Always remember that the code referring to “AFX” is actually MFC code.

Readers wishing an overview of C++ development with MFC are referred to books such as Microsoft Visual C++ Windows Applications by Example (by Stefan Bjornander).

The code for MPS is under Mps001.root, and the code for SR is under SR001.root. Under dev are support routines



## MPS

The organization of the MPS system is straightforward.

The MPS resides under Mps001.root/ Underneath the root folder is Mps001 which holds the solution, project, and source files.

The system builds into a Debug subfolder: There is also a Res (Resource) folder at this level..

The MPS001 SetupPackage folder was not examined.

## RS

The organization of the RS system is similar to MPS.

The RS resides under Rs001.root/ Underneath the root folder is Rs001 which holds the solution, project, and source files. There is also a RES folder which holds resources.

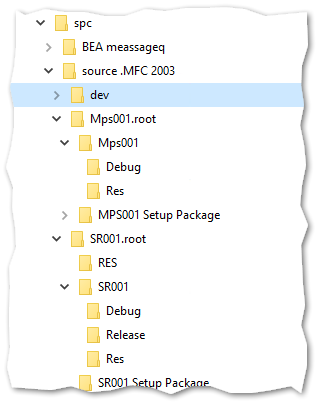
The system builds into one of the two subfolders: Release or Debug. There is also a Res (Resource) folder at this level, which is a copy of the RES folder.

The SR001 Setup Package folder was not examined.

## Support Code

Under dev are the programs for various support libraries. This includes:

1. CheckRes
2. Pps\_Lib
3. LibSpc
4. Spcres
5. CmTrace



## Check\_Res

Holds resource definitions. It is not clear if the resulting DLL (CheckRes70.DLL) is used anywhere.

## PPS\_LIB - LIB

PPS\_LIB loads the project NSCheck70, which builds into NSCheck70d.lib. This contains detailed information about the types of checks that are placed on slabs and coils, as well as the logic for rules violations. These appear to be no longer used, since the Menu Items

The file NsSql contains important SQL instructions for accessing constraints and chemistries.

The Resulting file is NSCheck70d.lib

## LibSpc - LIB

This library contains many SPC support routines for MPS and RS. This includes:

1. Methods for drawing the Coil and Slab Graphs
2. Support methods for the SQL interface with the SPC/Oracle systems
3. UI extensions for Windows controls.

The resulting file is LibSpcCs70\_1004d.lib

## SPCRES – DLL

A resource file, consisting of bitmaps, cursors, dialog references, string tables, etc.

The resulting file is SpcRes70d.DLL.

CMTrace – CPP and .H Files

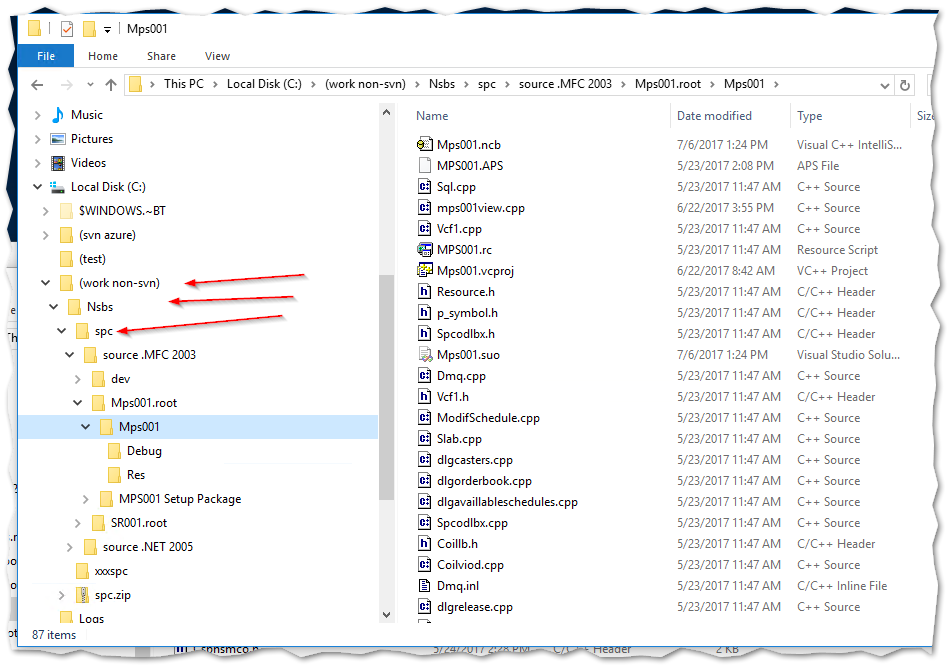
This is the TraceLog methods. Under its CM folder it holds trace.cpp and trace.h, that are included by any instrumented program.

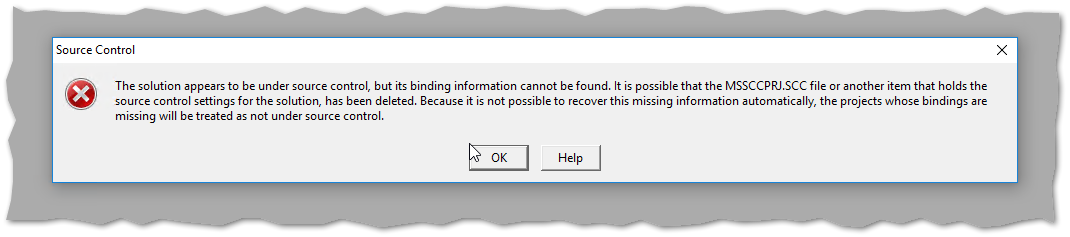
# Appendix – Getting the MPS and SR to Build

Used the VM Win10-VS2012, which is Windows 10, Version 1703, Build 15063.540.

Installed Visual Studio 2003 Pro from MSDN Account. Only installed the C++ portion (workflow).

Unzipped SPC.ZIP, and placed it under C:\NSBS



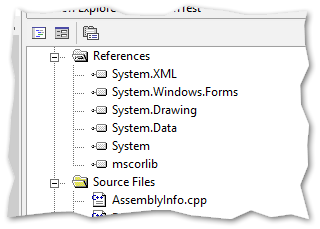


Ignored the SourceSafe errors.

Could not build because of a missing BEA Message Q file: p\_entry.h

Found the file in my archives (WDC2004) and was able to compile.

Could not link because of missing references, so built a stub WinForms C++ applications and used the references that were found there:



The link was then able to complete, but there were two missing DLLs at runtime:

Libct.dll and Libcs.dll.

These are apparently used by the database client (Sybase).

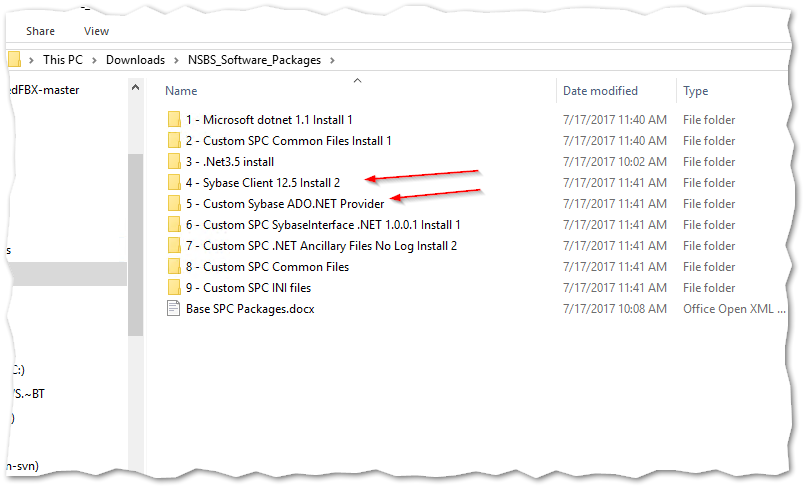
The search locations in the project > Settings are:

* ..\..\dev\LIBSPC.root\LIBSPC
* ..\..\dev\LIBSPC.root\LIBSPC\SybaseDebug
* ..\..\..\BEA MeassageQ\lib"
* ..\..\dev\Spcres.root\Spcres\Debug
* ..\..\dev\Pps\_lib.root\Pps\_lib\debug
* ..\..\dev\CheckRes.root\CheckRes\Debug

Note the annoying misspelling of MeasageQ (should be MessageQ)

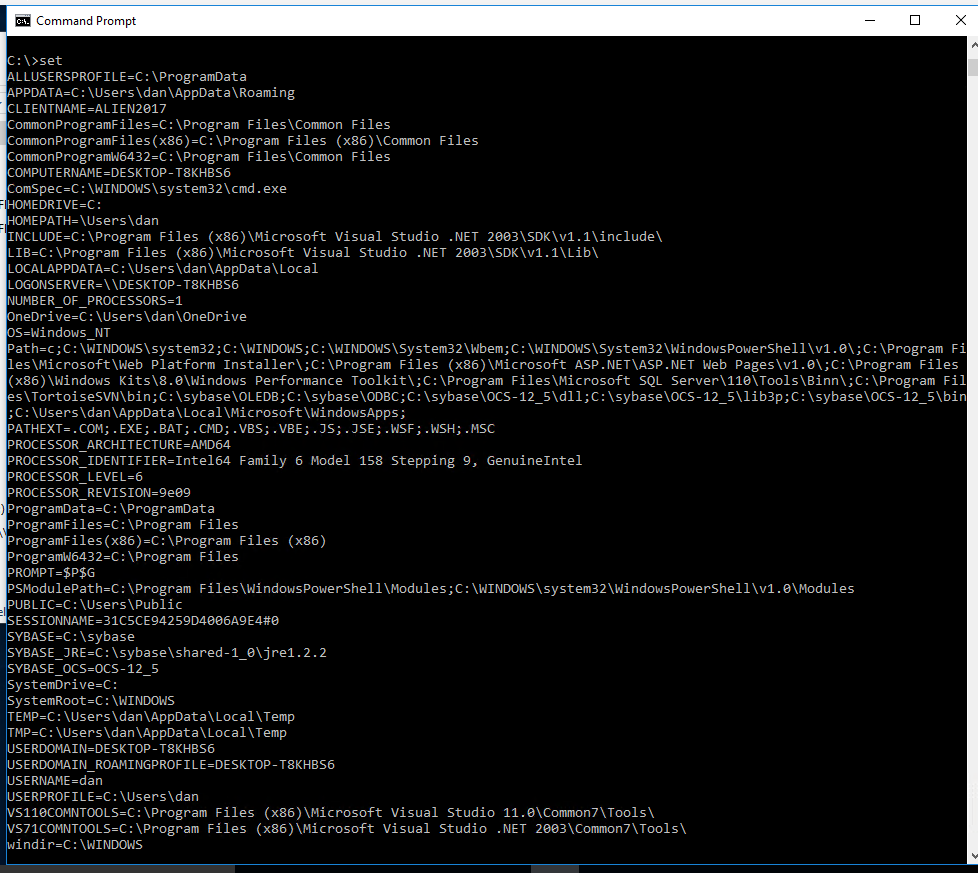
Needed the Sybase files libct.dll and libcs.dll.

Steve sent the Software\_Packages.zip file, which contained several installation kits, including the Sybase kits.



Under c:\sysbase\OCS012\_S\dll\debug and nondebug live the libc\*.dll files.

A reboot is necessary to pick up the environment variables:



After the reboot, the VS2003 program compiles and runs, and gives this prompt:

