

# Program Comprehension: A Strategy for the Bewildered

Frank DiIorio  
CodeCrafters, Inc.  
Chapel Hill, North Carolina  
Frank@CodeCraftersInc.com

## Background (2 of 3)

- You can take several approaches:
  - o Code for the sake of coding and looking busy, or you could ...
  - o Hope for a “Magical Day” → Stare at the program as you would a Magic Eye picture, hoping that its purpose and structure will eventually jump out at you. Or you could ..
  - o Remember what Hanna Arendt said – in a different context, probably, but still, it’s relevant ...

Background Setting Task Things To Do

## Background (1 of 3)

- The scenario ... you’re a
  - o SAS programmer who is
  - o new to a company and / or an application and
  - o must maintain, debug, or enhance an application
- You have good intentions, but ...
  - o there is **minimal** direction from above and **maximum** anxiety from within
  - o you need to know **how to start**. You need an organized approach to effectively understand the program.

Background Setting Task Things To Do

## Background (2 of 3)

*“Nothing can remain immense if it can be measured”*

- So, option #3 becomes ...
  - Calmly assess your situation and see what you need to learn before you start coding.
- Let’s assume you select the “assess and learn” option ...

Background Setting Task Things To Do

## This Paper (1 of 3)

- Objectives
  - o Help you identify groups of questions to ask, items to consider when you encounter a new program.
  - o Focus on how to acquire knowledge to work effectively on the programming task at hand (aka “cutting to the chase” or “finding the sweet spot” in the program)

Background Setting Task Things To Do

## This Paper (3 of 3)

- Organization
  - o Defining the problem
  - o Examining the program setting
  - o Determining the type of programming required
  - o Given the above, identify tools and techniques to facilitate program comprehension

Background Setting Task Things To Do

## This Paper (2 of 3)

- Caveats
  - o Many different ways to approach this
  - o No single way is “right”
  - o The goal is, in part, simply to make you aware of features of the programming environment to consider, as well as the types of programming you can undertake.
  - o The more questions and forethought, the better.  
**Don't just jump in and start coding!**

Background Setting Task Things To Do

## Review: Defining the Problem

- How do you come to understand the program?
- How do you acquire knowledge so you can work
  - o Efficiently?
  - o Effectively?
- There are two dimensions:
  - o The setting
  - o The task

Background Setting Task Things To Do

## The Setting

- What comprises the setting in which you work? What influences **how** and **what** you code?
- One way to organize this is:
  - o You
  - o Associates
  - o Corporate Setting
  - o Technical Environment
- All of these interact at some level. The art is determining which piece is most relevant to what you're doing.

Background Setting Task Things To Do

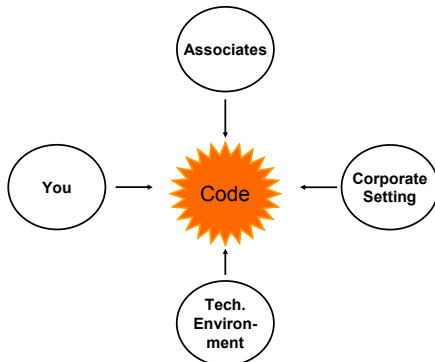
## The Setting – You

What do you know? How did you learn it?

- Domain Knowledge
  - o Your degree of familiarity with subject matter (can you distinguish between “good” and “bad” results?)
- Tool Knowledge
  - o Breadth and depth
- How You Learn
  - o Visual or text-based?
  - o Need the familiar to launch you to the unfamiliar?
  - o Pattern-recognition ability
  - o Willing to ask questions, ask for help?
  - o Learn by example or by formal instruction?

Background Setting Task Things To Do

## Setting - The Big Picture



Background Setting Task Things To Do

## The Setting – Associates

Who influences how you work and what you do?

- Internal Peers
  - o Programmers
  - o Analysts, management
  - o Multiple talents – programming, other technical, and domain expertise comprise the Program Team
- External Peers
  - o Mostly for technical, rather than domain, expertise
- Application Users
  - o Define (and change!) requirements
  - o Provide domain expertise

Background Setting Task Things To Do

## The Setting – Corporate Setting

What kind of environment does the program live in?

- Coding and Procedural Standards
  - o Internal (you, programming team, company-wide)
  - o External (mandated by client, regulatory, others)
- Communication of Needs
  - o Extent (established processes, or iterative, incremental specification)
  - o Means (formal, email, “oral history”)
  - o Learn how people communicate critical information

Background Setting Task Things To Do

## Setting Comments

- Everything’s connected and, ultimately, everything’s important.
- But, to understand the program and to know what to do, you must identify the features **most relevant** to your immediate needs.
- You must identify what you need to know.
- This is discussed in the next section.

Background Setting Task Things To Do

## The Setting – Technical Environment

What tools are available, and how is their use specified?

- Application Design
  - o The blueprint. What’s the level of specification?  
How is it presented?
- Capability and Features
  - o Available SAS (and other – utility and development) tools
  - o System (storage, network, etc.)

Background Setting Task Things To Do

## The Programming Task

- Why are you trying to understand the program in the first place? Different types of activities:
  - o Debugging
  - o Maintenance
  - o Enhancement
  - o Code Leverage and Reuse
- The type of task influences what Setting aspects become important
  - o In theory, everything is important and relevant, but gaining complete knowledge isn’t practical.

Background Setting Task Things To Do

## The Task – Debugging

- What is it?
  - Correcting code that produces erroneous or undesirable output
- Some Key Points from “Setting”
  - Syntax
  - Tool set
  - Design documents

Background Setting Task Things To Do

## The Task – Enhancement

- What Is It?
  - Adding non-routine functionality to an existing code base
- Some Key Points from “Setting”
  - End-user interaction
  - Your domain knowledge
  - Communication of needs

Background Setting Task Things To Do

## The Task – Maintenance

- What Is It?
  - Adding anticipated functionality to a system
- Some Key Points from “Setting”
  - End-user interaction
  - Knowledge of system architecture

Background Setting Task Things To Do

## The Task – Code Leverage and Reuse

- What Is It?
  - Integrating existing code into a new application
  - Writing generalized code for use by disparate applications
- Some Key Points from “Setting”
  - Documentation of architecture
  - Internal peers (programming team)
  - Tool capability

Background Setting Task Things To Do

## OK, Now What? Ten Things To Do

- Why 10? It just worked out that way ...
- Now that we know context and task, how do we get to work?
- Remember: the objective is to quickly gain an understanding of the **appropriate** pieces of the task environment, *to become as effective as possible as quickly as possible*.

Background Setting Task **Things To Do**

## 2. Start at the Right Altitude

- Altitude, aka level of detail
  - Context and detail differ by task: code leverage requires high altitude, Big Picture; debugging requires syntax, “where’s the missing END;” attention to detail
- Altitude changes as you work
  - Debugging may raise questions of design (low → high)
  - Reuse eventually gets down to syntax (high → low)

Background Setting Task **Things To Do**

## 1. Check Your Ego at the Door

- Domain expertise does not imply coding expertise
- Coding expertise does not imply analytical ability
- In general, expertise helps, but is never a guarantee of effective completion of the task at hand.
- Know when to ask for help

Background Setting Task **Things To Do**

## 3. Ask Questions, Provide Answers

- Ask questions, then code
- What’s program’s purpose? What, exactly, needs to be done?
- Task influences these questions
  - Debugging and leverage knowledge are acquired differently
  - Leverage/reuse: “what should the macro do?”
  - Debugging: “is it ok if the parameter is not upper-cased?”

Background Setting Task **Things To Do**

#### 4. Identify the “Piece Parts” (1 of 2)

- Only the simplest programs are stand-alone. Be aware of all possible pieces to the puzzle.
- Make a map (mental or otherwise) of the program and its context. Identify: format and macro libraries, startup files, macro variables, preceding and subsequent programs, etc. etc.
- Assess the impact of a change to Piece ‘x’ on other pieces (macros, programs, ...)

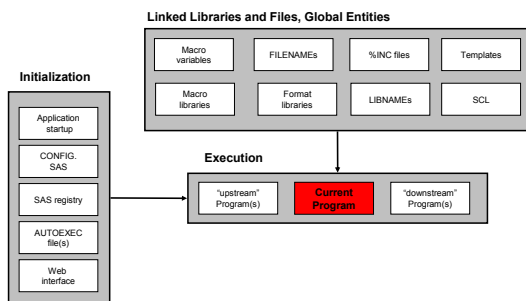
Background Setting Task **Things To Do**

#### 5. Understand Coding Styles

- Mandated and personal styles influence “look and feel” of a program
- They provide clues
  - o People solve routine tasks in (more or less) routine ways

Background Setting Task **Things To Do**

#### 4. Identify the “Piece Parts” (2 of 2)



Background Setting Task **Things To Do**

#### 6. Learn the Application’s History

- Consider the program’s history. Any changes to the:
  - o Data?
  - o Interface?
  - o Design “upstream” pieces?
- Any chronic “problem children”?
  - o Are they *really* fixed?

Background Setting Task **Things To Do**

## 7. Become Familiar with the Data

- The program won't make sense until you understand the data
- Issues of:
  - Granularity (“unit of observation”)
  - Missing value representation
- Review with:
  - VIEWTABLE, native viewer (for XLS, etc.)
  - Summary PROCs (MEANS, FREQ)

Background Setting Task **Things To Do**

## 9. Use Non-SAS Tools

- Viewing, analytic tools beyond those supplied with SAS
- Third-party (generalized or SAS-specific)
- Home-grown, for example:
  - Post-processing SAS Log
  - Macro cross-reference [see here](#)
  - Web page with organized links to program results
  - Customized display of Dictionary Table data

Background Setting Task **Things To Do**

## 8. Find Documentation

- In *any* form
- Comments in programs
- Specs
- Meta data, data dictionaries
- “Oral history”
- Data flow, other system diagrams

Background Setting Task **Things To Do**

## 10. Believe in the “Sweet Spot”

- Key to understanding what the program does is linked to *location*, not the *amount* of effort you expend.
- Look for patterns evolving over time – style, data problems, user enhancement requests – and the Sweet Spot becomes easier to find.

Background Setting Task **Things To Do**



## Thanks for Coming

Your comments are welcome and valued.  
Send email to: [fcd1@mindspring.com](mailto:fcd1@mindspring.com)  
or [Frank@CodeCraftersInc.com](mailto:Frank@CodeCraftersInc.com)

## Thanks for Coming

Your comments are welcome and valued.  
Send email to: [fcd1@mindspring.com](mailto:fcd1@mindspring.com)

