What’s up with the CF_SYLK and CF_DIF clipboard formats?

devblogs.microsoft.com/oldnewthing/20200226-00

February 26, 2020

Raymond Chen

A customer asked, “Please tell us about the `CF_SYLK` and `CF_DIF` clipboard formats and if you have information on the specific format to use.”

Okay, let’s start with the first part of the question: What are these things?

The Data Interchange Format is a file format introduced by VisiCalc. VisiCalc was the first electronic spreadsheet for personal computers, a landmark product which expanded the appeal of personal computers into the business world. It was arguably the first “killer app”, the program so compelling that it by itself served as justification for buying a computer. In 1983, VisiCalc ended up being overshadowed by Lotus 1-2-3 and quickly faded into history.

The SYLK format is a file format introduced by Microsoft Multiplan, one of the company’s early forays into integrated productivity software suites. Multiplan didn’t last for long, but it lasted long enough for Windows to add a clipboard format for it. Somewhat ironically, the Microsoft product that replaced Multiplan was Excel, which also served as one of the “killer apps” for that new operating system called Windows.

Now the second part of the question: Which one should we use?

That’s easy: Neither.

There’s no point adding support for these clipboard formats. You’d be interoperating with programs that nobody uses any more.

I asked why the customer was interested in these data formats in particular. After all, there are thousands of data formats out there. Why are they so interested in learning about DIF and SYLK? The customer seems to be running around looking for things, finding out what they are, and then deciding whether those things will help their program.

I can understand reaching DIF when starting with the requirement to support interoperability with VisiCalc: “We need to interoperate with VisiCalc. What data format does it use? Oh, DIF, thanks.”

But starting with DIF and asking what it is used for is going about things backward. Start with product requirements, and then identify the things that will help you achieve those requirements. “Will the DIF data format help us achieve the goal of X?” is a reasonable question. But “Tell me about the DIF data format” is an open-ended question that takes a lot of time to research, and it’s pointless to make someone do all that research only to say, “Oh, then I’m not interested.”

Its like saying, “The documentation says that if I have a document written in the Hittite language, I can tag it with `CF_HITTITE`. Where can I find documentation on the Hittite language?”

Why are you writing a document in Hittite? The last native speaker of Hittite died over 3000 years ago. There may be some academic scholars who can read Hittite, but if your intent is to communicate with those scholars, you probably should use a language they are more fluent in. If you’re going to send the document to, say, the Cuneiform Studies department at the University of Chicago, then you probably want to write your document in English.

What is your use case that led you to want to add DIF support? Do you have any customers who need DIF support? The last native speaker of DIF died in 1983.

Raymond Chen

Follow