Subj: Proposal for PAPI Profiles (and some PAPI V0.9 comments)
From: Tom Hastings and Ira McDonald
Date: 12/4/02
File: papi-profiles-and-comments-v2.doc

Add a new section, probably after Section 9, which defines what it means to conform to the PAPI specification, i.e., you have to claim conformance to either the Query Profile or the Submission Profile.

10 Conformance Profiles

This section defines two Conformance Profiles. The Job Submission Profile is a superset of the Query Profile. In order to claim conformance to this PAPI specification, any such claim must declare whether the implementation conforms to the requirements of the Query Profile or the Job Submission Profile. It is not sufficient to simply claim conformance to the PAPI specification.

10.1 Query Profile

The Query Profile is defined to provide querying functionality. A PAPI implementation conforming to the Query Profile must provide code for all functions defined in the PAPI and must support all of the definitions in the “papi.h” C header file. For each function defined in the PAPI specification, a conforming implementation must either perform the requested function or return the PAPI_OPERATION_NOT_SUPPORTED status code (see section 3.8). The PAPI_OPERATION_NOT_SUPPORTED status code indicates either (1) that the PAPI implementation doesn’t provide any support for the function, i.e., the function is stubbed out, or (2), the PAPI implementation does provide *code support* for the function, but the Printer or Print system selected by the application does not support the corresponding function.

Table 1 lists the functions and attributes that a PAPI implementation is REQUIRED to provide *code support* in order to claim conformance to the Query Profile. The blank entries are OPTIONAL for a PAPI implementation to support.

10.2 Job Submission Profile

The Job Submission Profile is defined to provide the job submission functionality and is a superset of the Querying Profile. Table 1 lists the functions and attributes that a PAPI implementation is REQUIRED to provide *code support* in order to claim conformance to the Job Submission Profile. The blank entries are OPTIONAL for a PAPI implementation to support.

Table 1 - PAPI Conformance Profiles

PAPI Functions and Attributes	Query Profile	Job Submission Profile
Section 3 Common Structures	all structures	all structures
Section 4 Service API	all functions REQUIRED[1]	all functions REQUIRED[1]
Section 5 Printer API:		
5.2 papiPrintersList	REQUIRED	REQUIRED
5.3 papiPrinterQuery	REQUIRED	REQUIRED
5.4 papiPrinterModify		
5.5 papiPrinterPause		
5.6 papiPrinterResume		
5.7 papiPrinterPurgeJobs		
5.8 papiPrinterListJobs	REQUIRED*	REQUIRED*
5.9 papiPrinterGetAttributeList	REQUIRED	REQUIRED
5.10 papiPrinterFree	REQUIRED	REQUIRED
5.11 papiPrinterListFree	REQUIRED	REQUIRED
Section 6 Attributes API	all functions REQUIRED	all functions REQUIRED
6.1 papiAttributeListAdd	REQUIRED	REQUIRED
6.2 papiAttributeListAddString	REQUIRED	REQUIRED
6.3 papiAttributeListAddInteger	REQUIRED	REQUIRED
6.4 papiAttributeListAddBoolean	REQUIRED	REQUIRED
6.5 papiAttributeListAddRange	REQUIRED	REQUIRED
6.6 papiAttributeListAddResolution	REQUIRED	REQUIRED
6.7 papiAttributeListAddDatetime	REQUIRED	REQUIRED
6.8 papiAttributeListAddCollection	REQUIRED	REQUIRED
6.9 papiAttributeDelete	REQUIRED	REQUIRED
6.10 papiAttributeListGetValue	REQUIRED	REQUIRED
6.11 papiAttributeListGetString	REQUIRED	REQUIRED
6.12 papiAttributeListGetInteger	REQUIRED	REQUIRED
6.13 papiAttributeListGetBoolean	REQUIRED	REQUIRED
6.14 papiAttributeListGetRange	REQUIRED	REQUIRED
6.15 papiAttributeListGetResolution	REQUIRED	REQUIRED
6.16 papiAttributeListGetDatetime	REQUIRED	REQUIRED
6.17 papiAttributeListGetCollection	REQUIRED	REQUIRED
6.18 papiAttributeListFree	REQUIRED	REQUIRED
6.19 papiAttributeListFind	REQUIRED	REQUIRED
6.20 papiAttributeListGetNext	REQUIRED	REQUIRED
6.21 papiAttributeListFromString		
6.22 papiAttributeListToString		

[1] See Ira's and Tom's comments to get rid of the SetServiceXxxx and GetServceXxxx and use the constructor only to set variables. Have a single Get function to return all values as a single structure, so that a service handle can be passed to a subroutine and the subroutine can Get the contents of the Service object. With these changes, *all* functions in Section 4 are REQUIRED for both Profiles.

PAPI Functions and Attributes	Query Profile	Job Submission Profile
Section 7 Job API:		
7.1 papiJobSubmit		REQUIRED
7.2 papiJobSubmitByReference		REQUIRED
7.3 papiJobValidate		
7.4 papiJobStreamOpen		REQUIRED
7.5 papiJobStreamWrite		REQUIRED
7.6 papiJobStreamClose		REQUIRED
7.7 papiJobQuery		REQUIRED*
7.8 papiJobModify		REQUIRED*
7.9 papiJobCancel		REQUIRED*
7.10 papiJobHold		REQUIRED*
7.11 papiJobRelease		REQUIRED*
7.12 papiJobRestart		REQUIRED*
7.13 papiJobGetAttributeList		REQUIRED
7.14 papiJobGetPrinterName		REQUIRED
7.15 papiJobGetId		REQUIRED
7.16 papiJobGetJobTicket		
7.17 papiJobFree		REQUIRED
7.18 papiJobListFree		REQUIRED
Section 8 Miscellaneous API:		REQUIRED
8.1 papiStatusString	REQUIRED	REQUIRED
Section 9 Attributes:		
9.1.1 job-ticket-formats-supported	REQUIRED	REQUIRED
9.1.2 media-margins	REQUIRED	REQUIRED
9.2 Required Job Attributes	REQUIRED	REQUIRED
9.3 Required Printer Attributes	REQUIRED	REQUIRED
9.4 IPP Attribute Type Mapping	REQUIRED	REQUIRED

* a conforming PAPI implementation MUST provide the code support for these functions and MUST NOT return the PAPI_OPERATION_NOT_SUPPORTED, unless the corresponding is not supported by the Printer or Print system or has been administratively disabled.

11 Comments on PAPI V0.9 while developing the profile

Ira and I have the following comments on PAPI V0.9 which we discovered while doing the profile work:

1. Section 2.2.4 Document and Appendix B. Do you want to add a reference to the IPP Document Object specification work in progress? Its available at ftp://ftp.pwg.org/pub/pwg/ipp/new_DOC/
2. Section 2.4 Globalization. The passive voice is used, so it isn't clear whether the section is talking about the application program that calls the PAPI or the PAPI implementation itself.

3. Section 3.3 Attributes and Values needs a reference for the data type codes to [RFC2911] section 4.1.
4. Section 3.8 (papi_status_t) needs a reference to [RFC2911] section 13 where the corresponding IPP status codes are defined.
5. It would be good to add a blank line in front of each new enum value assignment.
6. The two new status codes defined by PAPI (PAPI_BAD_REQUEST and PAPI_JOB_TICKET_NOT_SUPPORTED) need to be defined starting at a new place, since IPP will assign values beyond the last IPP server error (PAPI_PRINTER_IS_DEACTIVATED). So how about starting the PAPI defined status codes at 0x0600. We'll indicate this usage in the IPP code assignment table as well.
7. Section 4.5, uses the papi_encryption_t data type without a definition. Presumably, it is defined in the papi.h header file. How many other types are used in the spec without definition? Is this a problem?
8. The papi.h file needs to be stored with the spec and referenced from the spec, such like the IPP-Object-Attributes.pdf file is in Section 9. These files are part of the spec.
9. Get rid of all the SetServiceXxxx and GetServceXxxx and use the constructor only to set variables as in the current specification. If an application wants to change a Service attribute, such as user name or password, the application must construct a new service handle. Have a single Get function to return all values as a single structure, so that a service handle can be passed to a subroutine and the subroutine can Get the contents of the Service object. With these changes, *all* functions in Section 4 are REQUIRED for both Profiles.