Reserved Program Libraries

1. Introduction

This document describes some recommendations for those developers writing their own utility programs that, for convenience, are loaded from SYSRES ($CP) rather than a general-purpose application program unit ($P).

In addition to the conventions for program names and libraries this document also describes the naming conventions for 32-bit global symbols (in P.$SDLM0) and copy-book names in S.SYS32.

2. Command Libraries

As described in Technical Note IN191 the various P.$CMLB*x* command libraries were originally used for the following purposes:

P.$CMLB0	Command programs excluding $STARB customisation program
P.$CMLB1	Command programs
P.$CMLB2	Speedbase & 32-bit utilities
P.$CMLB3	Empty library, reserved for future use
P.$CMLB4	Stubs to SYSDEV, SYSKIT & SYSSBD

Since the release of GSM SP-6 this has been extended to:

P.$CMLB0	Command programs excluding $STARB customisation program
P.$CMLB1	Command programs
P.$CMLB2	Speedbase & 32-bit utilities
P.$CMLB3	**More 32-bit utilities**
P.$CMLB4	Stubs to SYSDEV, SYSKIT & SYSSBD

Some resellers have traditionally used P.$CMLB3 or P.$CMLB4 to hold their own $-commands. However these command libraries are now overwritten by the installation of a GSM Service Pack and the installation of the Global Development System: GSM SP-6, and later, install a fresh P.$CMLB3 on SYSRES; the installation of the Global Development System allocates a new P.$CMLB4 on SYSRES (to hold links to $-commands on SYSSBD).

We recognise the need for resellers to develop their own utility programs that reside on SYSRES (i.e. programs that are always load-able irrespective of the current $P assignment). Thus, the table of command libraries has been extended to:

P.$CMLB0	Command programs excluding $STARB customisation program
P.$CMLB1	Command programs
P.$CMLB2	Speedbase & 32-bit utilities
P.$CMLB3	More 32-bit utilities
P.$CMLB4	Stubs to SYSDEV, SYSKIT & SYSSBD
P.$CMLB5	**Reserved for private reseller/end-user command programs**
P.$CMLB6	**Reserved for future use by GSM - DO NOT USE**
P.$CMLB7	**Reserved for future use by GSM - DO NOT USE**
P.$CMLB8	**Reserved for future use by GSM - DO NOT USE**
P.$CMLB9	**Reserved for future use by GSM - DO NOT USE**

Note that command library P.$CMLB5 **is** already used internally at Global Business Systems to hold "internal-use" only commands that we have no intention of releasing externally. In this sense, P.$CMLB5 is already used for "private" utilities.

We guarantee that no GSM Service Pack or product installation will modify the P.$CMLB5 library, if one exists, on SYSRES.

3. $-commands and %-commands

Although the convention described in section 2 will prevent GSM command libraries clashing with the reserved reseller command library, there is still the potential for the **names** of GSM $-commands clashing with reseller $-commands. The names of new GSM utilities are allocated by a method that does **not** consider any external utilities therefore the possibility that a new GSM $-command will clash with your favourite "private" $-command does exist.

Those resellers who wish to register their own $-commands can do so by emailing Alan Underwood (alan@tissoft.co.uk) and we will **endeavour** (but **not** guarantee) to avoid clashes with future GSM utilities.

To avoid any possibility clashing program names resellers are **strongly** recommended to take advantage of a long-standing (albeit undocumented) feature in the program loader that loads programs that start with either a "$" or a "%" from SYSRES: **All reseller utility programs that reside on SYSRES (either as stand-alone programs or, more conveniently, in the P.$CMLB5 library) should start with the "%" symbol.** Note that GSM already uses this technique (invisibly) to load special handlers with a name of the form %.*xxxxxx* (e.g. %.W1D and %.W1S). Thus, the following convention should be followed:

$*xxxxxxx*	Reserved for GSM utilities

%.*xxxxxx*	Reserved for GSM special handlers
%*xxxxxxx*	Reserved for reseller utility programs

4. The Sting in the Tail

The $SDL32 compiler does not currently allow the program name specified by the PROGRAM or FRAME directive to start with a "%" symbol. For example:

FRAME $FRED	is allowed
FRAME %FRED	generates a compiler error

Until this restriction in the compiler has been removed resellers are recommended to compile with a temporary %-less name and use $F (or $LIB) to rename FRED as %FRED, for example, in a post-compilation stage in the development.

This can be done automatically, and very simply, by extending the $BUILD directives in the program source. For example:

```
FRAME FRED
*#BUILD          :$SDL32
*#OBJECT         :OBJ        :0
(other $BUILD directives to control the $SDL32 compilation)
*#RUN  :$F :OBJ :<CR> :REN :FRED :%FRED :<ESCAPE>
*#ENDBUILD
```

or, alternatively:

```
FRAME FRED
*#BUILD          :$SDL32
*#OBJECT         :OBJ        :0
(other $BUILD directives to control the $SDL32 compilation)
*#RUN  :$LIB :P.$CMLB5 :$CP :Y :COP :OBJ :FRED
*#RUN  :REN :FRED :%FRED :<CR> :END :<ESCAPE>
*#ENDBUILD
```

5. Summary of Program Files and Program Library Names

All non GSM "private" command programs should be named %*xxxxxxx* (where *xxxxxx* are any characters that do not start with ".") and placed in the P.$CMLB5 library on SYSRES.

6. DLM Entry Point Names and System Variables

The entry-points of all the externally available system subroutines within the system 32-bit DLM library, P.$SDLM0, include the "$" character as the last character in the global name (e.g. DS-DT$, GRAPH$ etc.) or start with "B$" (e.g. B$OPN). Other reserved, undocumented, global names, that should **not** be used by external developers (e.g. GR$AB1) will also include a "$" in the name (but not as the last character).

All externally available 32-bit System variables that are available via DLM's in the P.$SDLM0 DLM library will contain a "$" character as the first character of the name (e.g. $FUNC, $$AREA).

All 32-bit global symbols that include a "$" character in the name are reserved by Global. DO NOT USE THE "$" CHARACTER IN ANY 32-BIT GLOBAL SYMBOLS WITHIN APPLICATION SOFTWARE OTHERWISE SYMBOL NAME CLASHES MAY OCCUR.

If a global symbol within the P.$SDLM0 DLM library does not include a "$" character in the name then please report it as a bug (several "bugs" of this nature have recently been fixed in GSM SP-7).

7. The S.SYS32 Copy Library

We are endeavouring to keep the externally released S.SYS32 copy library in step with the newly released, and freshly documented, 32-bit sub-routines. Developers are encouraged to use this copy-library to define the control blocks required by 32-bit system subroutines.

In order to avoid copy-book name clashes with the existing copy-libraries that are currently used by developers all the copy-books defined in the S.SYS32 copy-library, **with the exception of one book**, will include the reserved "$" symbol in the name. The documentation for the sub-routine that requires a particular copy-book will explain how to expand the copy-book name correctly. This convention allows the creation of 125 (work it out yourself) "system" copy-books with names that include at least one reserved "$" character.

The only reserved $-less copy-book name is "FD". If you have already defined the FD copy-book for your own use and wish to use the S.SYS32 copy-library then the S.SYS32 copy-library must be defined as the last library in the compilation.