GXSLA$ - Set GX Dynamic Label Attribute

The GXSLA$ routine is used to set the attributes of a "Dynamic Label" in a window.

1. Invocation

To change the attributes of a Dynamic Label code:

```
      CALL GXSLA$ USING window index greyout attribute
```

or:

```
      CALL GXSLA$ USING window index greyout red green blue
```

where *window* is the Window Definition of the target window; *index* is the Dynamic Label index number (see below); *greyout* is a PIC 9(4) COMP field, or literal, set to either 0 (greyout) or 1 (normal display i.e. **not** greyed out); and *attribute* is the PIC 9(4) COMP, or literal, attribute number (between 0 and 64). In the 6 parameter call, which is only supported by GSM SP-16, or later, the *red*, *green* and *blue* parameters are PIC 9(4) COMP, or literal, colour weights (between 0 and 255).

To reset the attributes of a Dynamic Label code:

```
      CALL GXSLA$ USING wi index greyout
```

2. STOP Codes and Exception Conditions

The following STOP codes may be generated by GXSLA$:

STOP code	Description
15228	GXSLA$ has been called by an application that is not running on GX.
15229	The attribute number is invalid
15230	The label number is not in the range 1 to 99
15231	An attempt has been made to call GXSLA$ on an incompatible version of GX. The version of GX must be V2.8f, or later
15232	The window specified is not currently displayed
15233	The label index specified is not found

No exception conditions are returned by GXSLA$.

3. Programming Notes

GXSLA$ is only available when running on GX. Any attempt to use GXSLA$ on a non-GX terminal will result in a STOP code. The version of GX must be V2.8f, or later. The version of GSM must be GSM SP-9, or later.

A Dynamic Label is defined as a normal Speedbase label field with text of the following format:

"~I*nnText*"

where:

I	Dynamic Label indicator
nn	Dynamic Label index (i.e. 01 to 99, a leading zero is required for 01 to 09)
Text	Normal label text

The window containing the Dynamic Label **must** be displayed when GXSLA$ is called.

The Window Definition parameter **must** be the pointer to a Window Definition control block. It must not be a 2-character window-id. For example, the following code is correct:

```
CALL GXSLA$ USING W1 1 1 15
```

The following code is incorrect and will not have the desired effect:

```
CALL GXSLA$ USING "W1" 1 1 15
```

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

None.