

6 Character Classification and Utilities Module (chars.hhf)

The HLA CHARS module contains several procedures that classify and convert various character subtypes. Conversion routines include upper and lower case conversion. Classification routines include checking for alphabetic characters, numeric characters, whitespace characters, etc. This module works with ASCII characters in the range #0..#\$7F only. Though the functions accept 8-bit character values, non-ASCII characters generally do not get translated by the conversion routines and the predicate routines almost always return false for non-ASCII characters.

A Note About Thread Safety: The routines in this module are all thread safe.

Note about stack diagrams: To conserve space, this documentation does not include a stack diagram for any functions because none of the current "chars" functions pass data on the stack (that is, only a return address appears on the stack).

6.1 Conversion Functions

The conversion functions in the chars module convert (alphabetic) characters to lowercase and to uppercase.

```
chars.toUpper( c:byte ); @returns( "AL" );
```

This routine returns the character passed as a parameter in the AL register. If the character passed as a parameter was a lower case alphabetic character, this procedure converts it to upper case before returning it. Character values in the range #128..#255 are returned as-is; no conversion is done on those characters even if in some language they could be interpreted as alphabetic characters.

HLA high-level calling sequence examples:

```
chars.toUpper( charVar );
mov( al, uppercaseCharVar );
```

```
chars.toUpper( al );
// AL now contains the uppercase version.
```

```
mov( chars.toUpper( ch ), uppercaseCharVar ); // Char is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.toUpper;
mov( al, uppercaseCharVar );
```

```
// Char to convert is in AL
```

```
call chars.toUpper;
// Result char is left in AL
```

```
mov( ch, al );
call chars.toUpper;
mov( al, uppercaseCharVar );
```

```
chars.toLower( c:byte ); @returns( "AL" );
```

This routine returns the character passed as a parameter in the AL register. If the character passed as a parameter was an upper case alphabetic character, this procedure converts it to lower case before returning it.

Character values in the range #128..#255 are returned as-is; no conversion is done on those characters even if in some language they could be interpreted as alphabetic characters.

HLA high-level calling sequence examples:

```
chars.toLower( charVar );
mov( al, lowercaseCharVar );
```

```
chars.toLower( al );
// AL now contains the lowercase version.
```

```
mov( chars.toLower( ch ), lowercaseCharVar ); // Char is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.toLower;
mov( al, lowercaseCharVar );
```

```
// Char to convert is in AL
```

```
call chars.toLower;
// Result char is left in AL
```

```
mov( ch, al );
call chars.toLower;
mov( al, lowercaseCharVar );
```

6.2 Predicates (Tests)

The following functions test characters in the seven-bit ASCII character set. These functions produce undefined results for other character sets. Note: Although the "returns" value for each of these functions is "AL", in reality these functions all return the Boolean result zero-extended to 32 bits in EAX.

chars.isAlpha(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is an alphabetic character.

HLA high-level calling sequence examples:

```
chars.isAlpha( charVar );
mov( al, booleanVar );
```

```
chars.isAlpha( al );
// AL now contains the Boolean result.
```

```
mov( chars.isAlpha( ch ), booleanVar ); // Result is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.isAlpha;
mov( al, booleanVar );
```

```
// Char to convert is in AL

call chars.isAlpha;
// Result boolean is left in AL

mov( ch, al );
call chars.isAlpha;
mov( al, booleanVar );
```

chars.isUpper(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is an upper case alphabetic character.

HLA high-level calling sequence examples:

```
chars.isUpper( charVar );
mov( al, booleanVar );

chars.isUpper( al );
// AL now contains the Boolean result.

mov( chars.isUpper( ch ), booleanVar ); // Result is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.isUpper;
mov( al, booleanVar );

// Char to convert is in AL

call chars.isUpper;
// Result boolean is left in AL

mov( ch, al );
call chars.isUpper;
mov( al, booleanVar );
```

chars.isLower(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is a lower case alphabetic character.

HLA high-level calling sequence examples:

```
chars.isLower( charVar );
mov( al, booleanVar );

chars.isLower( al );
// AL now contains the Boolean result.

mov( chars.isLower( ch ), booleanVar ); // Result is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.isLower;
mov( al, booleanVar );

// Char to convert is in AL

call chars.isLower;
// Result boolean is left in AL

mov( ch, al );
call chars.isLower;
mov( al, booleanVar );
```

chars.isAlphaNum(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is an alphanumeric character.

HLA high-level calling sequence examples:

```
chars.isAlphaNum( charVar );
mov( al, booleanVar );

chars.isAlphaNum( al );
// AL now contains the Boolean result.

mov( chars.isAlphaNum( ch ), booleanVar ); // Result is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.isAlphaNum;
mov( al, booleanVar );

// Char to convert is in AL

call chars.isAlphaNum;
// Result boolean is left in AL

mov( ch, al );
call chars.isAlphaNum;
mov( al, booleanVar );
```

chars.isDigit(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is a decimal digit character.

HLA high-level calling sequence examples:

```
chars.isDigit( charVar );
mov( al, booleanVar );
```

```

chars.isDigit( al );
// AL now contains the Boolean result.

mov( chars.isDigit( ch ), booleanVar ); // Result is left in AL.

```

HLA low-level calling sequence examples:

```

mov( charVar, al );
call chars.isDigit;
mov( al, booleanVar );

// Char to convert is in AL

call chars.isDigit;
// Result boolean is left in AL

mov( ch, al );
call chars.isDigit;
mov( al, booleanVar );

```

chars.isXDigit(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is a hexadecimal digit character.

HLA high-level calling sequence examples:

```

chars.isXDigit( charVar );
mov( al, booleanVar );

chars.isXDigit( al );
// AL now contains the Boolean result.

mov( chars.isXDigit( ch ), booleanVar ); // Result is left in AL.

```

HLA low-level calling sequence examples:

```

mov( charVar, al );
call chars.isXDigit;
mov( al, booleanVar );

// Char to convert is in AL

call chars.isXDigit;
// Result boolean is left in AL

mov( ch, al );
call chars.isXDigit;
mov( al, booleanVar );

```

chars.isGraphic(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is a printable character or a space (this excludes control characters; also, this function only applies to ASCII characters).

HLA high-level calling sequence examples:

```
chars.isGraphic( charVar );
mov( al, booleanVar );
```

```
chars.isGraphic( al );
// AL now contains the Boolean result.
```

```
mov( chars.isGraphic( ch ), booleanVar ); // Result is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.isGraphic;
mov( al, booleanVar );
```

```
// Char to convert is in AL
```

```
call chars.isGraphic;
// Result boolean is left in AL
```

```
mov( ch, al );
call chars.isGraphic;
mov( al, booleanVar );
```

chars.isSpace(c:byte); @returns("AL");

This routine returns true in the AL register if the parameter is a white space character. A white space character is a space, carriage return, linefeed, or tab character.

HLA high-level calling sequence examples:

```
chars.isSpace( charVar );
mov( al, booleanVar );
```

```
chars.isSpace( al );
// AL now contains the Boolean result.
```

```
mov( chars.isSpace( ch ), booleanVar ); // Result is left in AL.
```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.isSpace;
mov( al, booleanVar );
```

```
// Char to convert is in AL
```

```

call chars.isSpace;
// Result boolean is left in AL

mov( ch, al );
call chars.isSpace;
mov( al, booleanVar );

```

chars.isASCII(c:byte); @returns("AL");

This routine returns true in AL if the parameter byte is an ASCII character (value in the range \$0..\$7F).

HLA high-level calling sequence examples:

```

chars.isASCII( charVar );
mov( al, booleanVar );

```

```

chars.isASCII( al );
// AL now contains the Boolean result.

```

```

mov( chars.isASCII( ch ), booleanVar ); // Result is left in AL.

```

HLA low-level calling sequence examples:

```

mov( charVar, al );
call chars.isASCII;
mov( al, booleanVar );

// Char to convert is in AL

call chars.isASCII;
// Result boolean is left in AL

mov( ch, al );
call chars.isASCII;
mov( al, booleanVar );

```

chars.isCtrl(c:byte); @returns("AL");

This function returns true in AL if the parameter is a control character (\$0..\$1F or \$7F).

HLA high-level calling sequence examples:

```

chars.isCtrl( charVar );
mov( al, booleanVar );

```

```

chars.isCtrl( al );
// AL now contains the Boolean result.

```

```

mov( chars.isCtrl( ch ), booleanVar ); // Result is left in AL.

```

HLA low-level calling sequence examples:

```
mov( charVar, al );
call chars.isCtrl;
mov( al, booleanVar );

// Char to convert is in AL

call chars.isCtrl;
// Result boolean is left in AL

mov( ch, al );
call chars.isCtrl;
mov( al, booleanVar );
```