

LibCommon_TextStrings.ttcn_etsi(NEW)	LibCommon_TextStrings_oma.ttcn(OLD)
/**	/*
* @author ETSI	* @author ETSI STF 276
* @version \$Id: LibCommon_TextStrings.ttcn 19 2008-01-31 11:28:24Z schulzs \$	* @version \$Id\$
* @desc A collection of text string type and value definitions which * may be useful in the implementation of any TTCN-3 test * suite. "Text string" refers to TTCN-3 charstring and universal * charstring types.	* @desc A collection of text string type and value definitions which * may be useful in the implementation of any TTCN-3 test * suite. "Text string" refers to TTCN-3 charstring and universal * charstring types.
* @remark End users should be aware that any changes made to the in * definitions this module may be overwritten in future releases. * End users are encouraged to contact the distributors of this * module regarding their modifications or additions so that future * updates will include your changes.	* @remark End users should be aware that any changes made to the in * definitions this module may be overwritten in future releases. * End users are encouraged to contact the distributors of this * module regarding their modifications or additions so that future * updates will include your changes.
*/	*/
module LibCommon_TextStrings language "TTCN-3:2005" {	module LibCommon_TextStrings {
/**	/*

LibCommon_TextStrings.ttcn_etsi(NEW)	LibCommon_TextStrings_oma.ttcn(OLD)
<pre> * @desc These constants can be used to add special characters into *       TTCN-3 text strings by using the concatenation operator. *       Example use: *       var charstring v_text := "Hi!" &amp; c_CRLF &amp; "Hello!"; */ group usefulConstants { </pre>	<pre> * @desc These constants can be used to add special characters into *       TTCN-3 text strings by using the concatenation operator. *       Example use: *       var charstring v_text := "Hi!" &amp; c_CRLF &amp; "Hello!"; */ group usefulConstants { </pre>
const charstring c_NUL := oct2char('00'O);	const charstring c_NUL := oct2str('00'O);
const charstring c_SOH := oct2char('01'O);	const charstring c_SOH := oct2str('01'O);
const charstring c_STX := oct2char('02'O);	const charstring c_STX := oct2str('02'O);
const charstring c_ETX := oct2char('03'O);	const charstring c_ETX := oct2str('03'O);
const charstring c_EOT := oct2char('04'O);	const charstring c_EOT := oct2str('04'O);
const charstring c_ENQ := oct2char('05'O);	const charstring c_ENQ := oct2str('05'O);
const charstring c_ACK := oct2char('06'O);	const charstring c_ACK := oct2str('06'O);
const charstring c_BEL := oct2char('07'O);	const charstring c_BEL := oct2str('07'O);
const charstring c_BS := oct2char('08'O);	const charstring c_BS := oct2str('08'O);
const charstring c_TAB := oct2char('09'O);	const charstring c_TAB := oct2str('09'O);

LibCommon_TextStrings.ttcn_etsi(NEW)	LibCommon_TextStrings_oma.ttcn(OLD)
const charstring c_LF := oct2char('0A'O);	const charstring c_LF := oct2str('0A'O);
const charstring c_VT := oct2char('0B'O);	const charstring c_VT := oct2str('0B'O);
const charstring c_FF := oct2char('0C'O);	const charstring c_FF := oct2str('0C'O);
const charstring c_CR := oct2char('0D'O);	const charstring c_CR := oct2str('0D'O);
const charstring c_SO := oct2char('0E'O);	const charstring c_SO := oct2str('0E'O);
const charstring c_SI := oct2char('0F'O);	const charstring c_SI := oct2str('0F'O);
const charstring c_DLE := oct2char('10'O);	const charstring c_DLE := oct2str('10'O);
const charstring c_DC1 := oct2char('11'O);	const charstring c_DC1 := oct2str('11'O);
const charstring c_DC2 := oct2char('12'O);	const charstring c_DC2 := oct2str('12'O);
const charstring c_DC3 := oct2char('13'O);	const charstring c_DC3 := oct2str('13'O);
const charstring c_DC4 := oct2char('14'O);	const charstring c_DC4 := oct2str('14'O);
const charstring c_NAK := oct2char('15'O);	const charstring c_NAK := oct2str('15'O);
const charstring c_SYN := oct2char('16'O);	const charstring c_SYN := oct2str('16'O);
const charstring c_ETB := oct2char('17'O);	const charstring c_ETB := oct2str('17'O);
const charstring c_CAN := oct2char('18'O);	const charstring c_CAN := oct2str('18'O);
const charstring c_EM := oct2char('19'O);	const charstring c_EM := oct2str('19'O);
const charstring c_SUB := oct2char('1A'O);	const charstring c_SUB := oct2str('1A'O);

LibCommon_TextStrings.ttcn_etsi(NEW)	LibCommon_TextStrings_oma.ttcn(OLD)
const charstring c_ESC := oct2char('1B'O);	const charstring c_ESC := oct2str('1B'O);
const charstring c_FS := oct2char('1C'O);	const charstring c_FS := oct2str('1C'O);
const charstring c_GS := oct2char('1D'O);	const charstring c_GS := oct2str('1D'O);
const charstring c_RS := oct2char('1E'O);	const charstring c_RS := oct2str('1E'O);
const charstring c_US := oct2char('1F'O);	const charstring c_US := oct2str('1F'O);
const charstring c_DEL := oct2char('7F'O);	const charstring c_DEL := oct2str('7F'O);
const charstring c_CRLF := oct2char('0D'O) & oct2char('0A'O);	const charstring c_CRLF := oct2str('0D'O) & oct2str('0A'O);
} // end group usefulConstants	} // end group usefulConstants
/**	/*
* @remark Number in name indicates string length in number of	* @remark Number in name indicates string length in number of
*     _characters_	*     _characters_
*/	*/
group textStringSubTypes {	group textStringSubTypes {

LibCommon_TextStrings.ttcn_etsi(NEW)		LibCommon_TextStrings_oma.ttcn(OLD)	
type charstring String1	length(1) with { encode "length(1)"};	type charstring String1	length(1) with { encode "length(1)"};
type charstring String2	length(2) with { encode "length(2)"};	type charstring String2	length(2) with { encode "length(2)"};
type charstring String3	length(3) with { encode "length(3)"};	type charstring String3	length(3) with { encode "length(3)"};
type charstring String4	length(4) with { encode "length(4)"};	type charstring String4	length(4) with { encode "length(4)"};
type charstring String5	length(5) with { encode "length(5)"};	type charstring String5	length(5) with { encode "length(5)"};
type charstring String6	length(6) with { encode "length(6)"};	type charstring String6	length(6) with { encode "length(6)"};
type charstring String7	length(7) with { encode "length(7)"};	type charstring String7	length(7) with { encode "length(7)"};
type charstring String8	length(8) with { encode "length(8)"};	type charstring String8	length(8) with { encode "length(8)"};
type charstring String9	length(9) with { encode "length(9)"};	type charstring String9	length(9) with { encode "length(9)"};
type charstring String10	length(10) with { encode "length(10)"};	type charstring String10	length(10) with { encode "length(10)"};
type charstring String11	length(11) with { encode "length(11)"};	type charstring String11	length(11) with { encode "length(11)"};
type charstring String12	length(12) with { encode "length(12)"};	type charstring String12	length(12) with { encode "length(12)"};
type charstring String13	length(13) with { encode "length(13)"};	type charstring String13	length(13) with { encode "length(13)"};
type charstring String14	length(14) with { encode "length(14)"};	type charstring String14	length(14) with { encode "length(14)"};
type charstring String15	length(15) with { encode "length(15)"};	type charstring String15	length(15) with { encode "length(15)"};
type charstring String16	length(16) with { encode "length(16)"};	type charstring String16	length(16) with { encode "length(16)"};

LibCommon_TextStrings.ttcn_etsi(NEW)	LibCommon_TextStrings_oma.ttcn(OLD)
type charstring String1To63 length(1..63) with {encode "length(1..63)"};	type charstring String1To63 length(1..63) with {encode "length(1..63)"};
type charstring String1To64 length(1..64) with {encode "length(1..64)"};	type charstring String1To64 length(1..64) with {encode "length(1..64)"};
type charstring String1To127 length(1..127) with {encode "length(1..127)"};	type charstring String1To127 length(1..127) with {encode "length(1..127)"};
type charstring String1To128 length(1..128) with {encode "length(1..128)"};	type charstring String1To128 length(1..128) with {encode "length(1..128)"};
type charstring String1to255 length(1..255) with {encode "length(1..255)"};	type charstring String1to255 length(1..255) with {encode "length(1..255)"};
type charstring String5to253 length (5..253) with {encode "length(5..253)"};	
} // end stringSubTypes	} // end stringSubTypes
group usefulTextStringTypes {	group usefulTextStringTypes {
type universal charstring UnicodeText;	type universal charstring UnicodeText;
type universal charstring UnicodeText1to255 length(1..255) with {encode "length(1..255)"};	type universal charstring UnicodeText1to255 length(1..255) with {encode "length(1..255)"};
/**	type charstring AlphaNum ("0".."9","a".."z","A".."Z");

LibCommon_TextStrings.ttcn_etsi(NEW)	LibCommon_TextStrings_oma.ttcn(OLD)
* @desc Subtyping can not be used in this type definition to ensure	type AlphaNum AlphaNum2 length(2) with { encode "length(2)"};
* values of this type are really alphanumeric.	type AlphaNum AlphaNum1To32 length(1..32) with {encode "length(1..32)"};
* Instead either codecs or a template have	
* to be used for this purpose. The type is kept here to ensure	
* backward compatibility.	
* @see LibCommon_TextStrings.mw_isAlphaNum	
*/	
type charstring AlphaNum with { encode "alphanumeric"};	
type AlphaNum AlphaNum2 length(2) with { encode "alphanumeric;length(2)"};	
type AlphaNum AlphaNum1To32 length(1..32) with {encode "alphanumeric;length(1..32)"};	
} // end group usefulTextStringTypes	} // end group usefulTextStringTypes
group usefulTextStringTemplates {	

LibCommon_TextStrings.ttcn_etsi(NEW)	LibCommon_TextStrings_oma.ttcn(OLD)
template charstring mw_isAlphaNum := pattern "([0-9][a-z][A-Z])+";	
} // end group usefulTextStringTemplates	
} // end module LibCommon_TextStrings	} // end module LibCommon_TextStrings