

## INSTALLATION AND OVERVIEW OF powerEXT Core Version 4.1

\*\*\*\*\*

**IMPORTANT NOTE: powerEXT 4.0 has been changed in many subprocedure interfaces in order to support variable lengths and I/O for up to 65535 bytes and it is therefor not compatible with earlier Core versions and all programs using it must be recompiled.**

\*\*\*\*\*

PLEASE NOTE: powerEXT Core 4.x is no longer part of the powerEXT Application Framework, but is now a prerequisite for powerEXT AF.

### Includes:

- \* Special version of CGIDEV2
  - UTF8/16 file support
  - Responce Object Support
  - Variable length 65535 support
- \* powerEXT Core API: HTTP connector & Productivity Services
- \* sample programs

### Documentation:

The powerEXT API is documented on (case sensitive)  
[http://powerEXT.com/pextdoc\\_CGI.htm](http://powerEXT.com/pextdoc_CGI.htm)

### Installation:

The installation procedure is simple and requires  
i5/OS V5R2 & WDS:

**\*\* the installation should be run under an \*\***  
**\*\* user profile with \*ALLOBJ authority \*\***

1. Download the savf to your PC
2. FTP it to a savf on your iSeries
3. Restore the library PEXTCD2 from it
4. Run CALL PEXTCD2/PXCRTCD2
  - converts all source files to your CCSID and  
recompiles all programs to current i5/OS version.
5. include PEXTCD2 in your library list

**powerEXT Core**

- MIT licensed, for the CGIDEV2 part please see the license notes included in CGIDEV2
- supports basic client centric RIA development
- features and methods on [http://www.powerEXT.com/pextdoc\\_CGI.htm](http://www.powerEXT.com/pextdoc_CGI.htm)
- lowest OS i release V5R2
- builds on an enhanced CGIDEV2 and uses the following subprocedures from CGIDEV2:

- clrHtmlBuffer
- getEnv
- zhbGetInput
- zhbGetVarCnt
- zhbGetVar
- getHtmlIfs
- updHtmlVar2
- wrtSection
- wrtNoSection
- wrtHtmlToStmf (enhanced by Scott to support UTF-8)
- zhbBufAddr (added to CGIDEV2 as part of the powerEXT project)
- zhbBufSize (added to CGIDEV2 as part of the powerEXT project)

### **Basic HTTP I/O support and ILOB management (CGIDEV2 connect and enhance)**

initiate:

clearSrvPgm()	Initiates the serviceprogram
---------------	------------------------------

http input:

getExtInput()	Get input (CGIDEV2 /copy qrpglesrc,prolog3)
getExtInputRaw()	Get QSTDIN raw into a store, returns store no.
getExtEnvVar()	Get Environment Variable (CGIDEV2 GetEnv)
getExtVar()	Get Form Variable (CGIDEV2 zhbGetInput)
getExtVarNum()	Get Numeric Form Variable (CGIDEV2 zhbGetInput)
inputAddr()	Returns %addr on input
inputSize()	Returns %size on input

build ILOB/output buffer:

setContent()	Adds a content type (default text/html)
setExtVar()	Update a HTML variable (CGIDEV2 updHTMLvar)
echo()	Append a textstring to the out-buffer (CGIDEV2 wrtNoSection)
echoAsp()	Adds a template section with ASP delimiters (CGIDEV2 wrtSection)
echoCgi()	Adds a template section with CGIDEV2 special delimiters (CGIDEV2 wrtSection)
echoCgiDev2()	Adds a template section with CGIDEV2 standard or optional delimiters (CGIDEV2 wrtSection)

output ILOB/output buffer:

echoToClient()	Sends output buffer to browser (CGIDEV2 wrtSection('*fini')
echoToStmf()	Sends output buffer to stmf
bufAddr()	Returns %addr on buffer
bufSize()	Returns %size on buffer

others:

echoBinToClient()	Sends a binary stmf to browser
-------------------	--------------------------------

### **Dynamic Memory Stores (similar to userspaces)**

storeNew()	Returns next free store number (1-1000)
storeInz()	Initialize a store
storeAppend()	Append data to a store by %addr/%size
storeAppendText()	Append textstring to a store
storeAddr()	Returns %addr on a store
storeSize()	Returns %size on a store
storeFree()	Free storage and storenumber
storeFromStmf()	Reads data from stmf to a store
storeToStmf()	Writes data from a store to a stmf
storeToField()	Returns up to 65535 bytes from store to a field

### **Structured node (HTML, XML, JSON, CSV)**

htmlNode() / htmlEndNode()	HTML Node support, encodes automatically
xmlNode() / xmlEndNode()	XML Node support, encodes automatically
jsonNode() / jsonEndNode()	JSON Node Support, encodes automatically
csvNode() / csvEndNode()	CSV Node/Row support, encodes automatically
csvDelimiter()	Sets the CSV delimiter (,;/)

### **Encode/decode support**

encodeHTML()	Encodes special char. in HTML
encodeXML()/decodeXML()	Encodes/decodes special char. in XML
encodeJSON()/decodeJSON()	Encodes/decodes special char. in JSON
encodeCSV()	Encodes special char. in CSV strings
encodeJAVA()	Encodes special char. in JAVA strings
encodeRPG	Encodes special char. in RPG strings

### **XML Reader, JSON/CSV to XML converters**

xmlFromStmf()	Reads a text based file into a special buffer
-xmlAddr()	Returns the %addr of the special buffer
-xmlSize()	Returns the %size of the special buffer
xmlToStmf()	Sends output buffer (store 0) to a stmf
xmlReaderInz()	Initialize the xmlReader

xmlReaderCase()	Sets the reader to return nodes in upper/lower/mixed case
xmlReader()	Reads next XML element or attribute
-xmlGetNodeType()	Returns Nodetype on current element
-xmlGetDepth()	Returns Three Depth of current element
-xmlGetNode()	Returns Node Name of current element
-xmlGetXPath()	Returns XPath or part of XPath of current element
-xmlGetXPathJson()	Returns XPath or part of XPath in JSON converted file
-xmlGetAttr()	Returns Current Attribute Name if the reader is in attribute section
-xmlGetData()	Returns Data Part of current element
-xmlGetDataNum()	Returns Numeric Part of current element
-xmlAddrOuter()	Returns the outer %addr of current element
-xmlSizeOuter()	Returns the %size of current element
-xmlAddrInner()	Returns the inner %addr of current element
-xmlSizeInner()	Returns the inner %size of current element
-jsonToXML()	Converts JSON to XML
-csvToXML()	Converts CSV to XML

#### **Internal code generation tabulation support**

setGlobalTab()	Set tabulation off/on in echo and nodesupport
setGlobalNL()	Set New Line off/on in echo and nodesupport
tab()	Increases tabulation counter by 1
tabEnd()	Decreases tabulation counter by 1
tabClear()	Clears tabulation counter
NL()	Returns a string with a NL and current numbers of tabs

#### **Misc.**

timeStamp()	Returns a timestamp with all 6 microseconds
uniqueKey()	Returns a 32 bytes unique key
isoFromNum()	Converts a Numeric Date to a ISO date string
isoToNum()	Converts a ISO Date string to a Numeric Date
numMaxValue()	Returns a 99999 string based of field length and dec pos
qCmd()	Processes QCMDEXE
convCCS()	Converts CCSid in a string