getServiceContextValue

Syntax	`aString = getServiceContextValue(protocol, parameter)`		
Semantics	Returns a **String** with the value of the chosen service context parameter related to the current request.		
Substitutables	`protocol`	A **String** specifying the protocol, of which a parameter is requested.	
	`parameter`	A **String** specifying the name of the requested parameter.	

Examples	`set ip = getServiceContextValue("tcpip"," remoteIPAddress");`
	`set agent = getServiceContextValue("http","user-agent");`
	
	`//currentServiceDirectory = the home of the current configuration, e.g.: D:\E2E_BRIDGE_DATA\bridge_Test`
	`set currentServiceDirectory = getServiceContextValue ("SystemEnvironment","INSTANCE_HOME");`		

Supported protocols

Supported protocols are dependant on the service type. See function `getServiceContext()` for more details.

Supported parameters

Protocol	Parameter	Description
TCPIP	**RemoteIPAddress**	IP address of host that initiated the request. If a proxy is used, this is the IP of the proxy server.
	RemotePort	Port number, from which the request was initiated.
	RemoteHostName	Name of the host that initiated the request. If not resolvable, the IP address is returned.
	LocalIPAddress	IP address of the machine this service runs on.
	LocalPort	Port number of this service.
	LocalHostName	Name of the host this service runs on. If not resolvable, the IP address is returned.
HTTP	**Method**	HTTP method used to invoke this service. Usually, it returns **POST**.
	URI	URI called to invoke this service.
	Body	Complete content of the HTTP request invoking this service.
	<header>	Any other name will return the value of the HTTP header of the same name, if it exists.
SOAP	**Header**	Complete source of the SOAP header sent to invoke this service.
	Body	Complete source of the SOAP body sent to invoke this service.

SystemEnvironment	<environment variable name>	Value of the operating system environment variable. This call returns a value for all service types. Note: the accessible environment variables can be inspected in the **start log** of the current service (see Server User Guide for more details). ⓘ PAS uses some standard environment variables which are all uppercase on Linux. On Windows, they are all in lowercase, but unlike Linux, Windows is not case-sensitive. Therefore, we recommend to provide all variable names in uppercase, e.g. SERVER_HOSTNAME since this works in both environments.
System	**Host**	Local host name. This call returns a value for all service types.
	IP	IP address of the local host. This call returns a value for all service types.
	User	The operating system user who is executing the current service. This user has nothing to do with the external user who might invoke a service. Thus, this call returns a value for all service types.