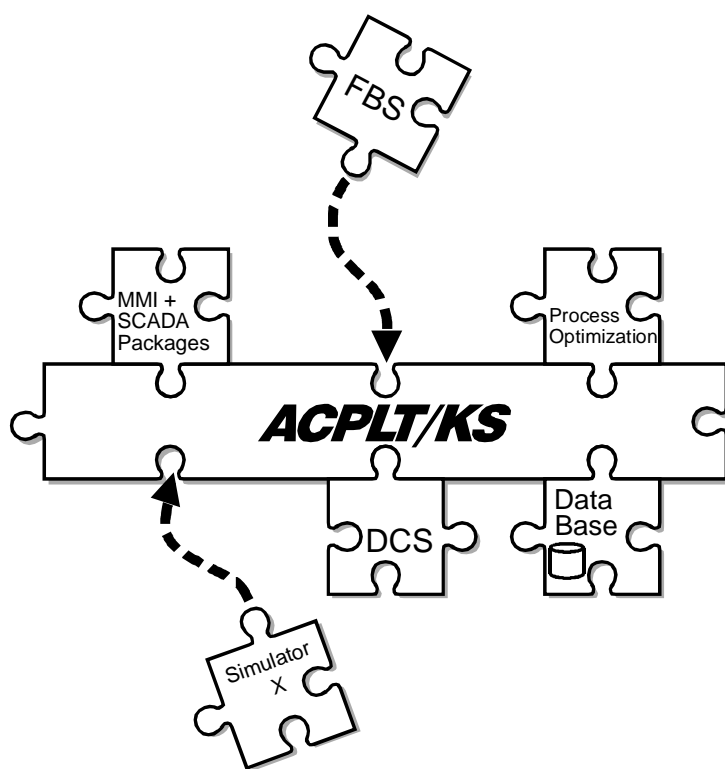


# ***ACPLT/KS***

## Technologiepapier Nr. 6: Implementierungshinweise



## Inhalt

1 Einleitung .....	3
2 Das KS-Kernprotokoll.....	3
2.1 RPC-Schicht .....	3
2.2 Dienstegruppen.....	3
3 Der Objektbaum .....	4
3.1 Manager und Server .....	4
3.2 Protokollerweiterungen .....	5
3.3 Managerspezifische Objekte .....	6
4 KS-Fehlercodes .....	7
5 AuA – Abkürzungen und Akronyme.....	9
6 Literaturverzeichnis .....	9

## 1 Einleitung

Das Kommunikationssystem ACPLT/KS (im folgenden auch „KS“ genannt) verbindet im Bereich von Anwendungen der Prozeß- und Betriebsführung rechnergestützte Werkzeuge untereinander und mit Prozeßleitsystemen. Dieses Technologiepapier beschreibt Implementierungsaspekte der verschiedenen ACPLT/KS-Teilnehmer.

## 2 Das KS-Kernprotokoll

### 2.1 RPC-Schicht

Das ACPLT/KS-Nachrichtenprotokoll besitzt die (weltweit eindeutige) RPC-Programmnummer 0x49678. Die aktuelle Versionsnummer ist 1. Dementsprechend sieht die Deklaration der RPC-Diensteschnittstelle in der rpcgen-Syntax [1] wie folgt aus:

```
program KS_PROG {
    version KS_VERS {
        ...
    } = 1;
} = 0x49678;
```

### 2.2 Dienstgruppen

Das KS-Kernprotokoll teilt sich in (aktuell) vier Dienstgruppen auf. Jede Dienstgruppe enthält einen zusammengehörenden Satz von Diensten (Services). Die Implementierung bestimmter Dienstgruppen ist Pflicht („ERF“ = erforderlich), wohingegen andere Dienstgruppen optional (= „OPT“) sind. Die folgende Übersicht legt weiterhin fest, für welchen Typ von KS-Teilnehmer (Server, Manager) eine Dienstgruppe implementiert sein muß beziehungsweise gegebenenfalls optional implementiert werden kann. Der Name der Dienstgruppe stellt zugleich eine Kennung dar, die im Variablenobjekt "/vendor/extensions/ks\_core/services" (siehe Abschnitt 3.2) in der Liste der verfügbaren Dienstgruppen abgelegt ist.

Dienstgruppe	Server	Manager
ObjDict	ERF	ERF

Die Dienstgruppe "ObjDict" dient dazu, durch den Objektbaum zu „wandern“, um so die verfügbaren Kommunikationsobjekte in einem Server (oder Manager) aufzuzählen. Es sind die nachfolgend aufgeführten Services verfügbar (RPC-Prozedurnummer in Klammern):

KS\_GETPP (0x00000001)

Dienstgruppe	Server	Manager
Var	ERF	ERF

Die Dienstgruppe "Var" umfaßt Funktionen zum Lesen und Schreiben von Variablen. Es sind die nachfolgend aufgeführten Services verfügbar (RPC-Prozedurnummer in Klammern):

KS\_GETVAR (0x00000101)

KS\_SETVAR (0x00000102)

Dienstgruppe	Server	Manager
DataExchg	OPT	—

Die Dienstgruppe "DataExchg" stellt einen Mechanismus zur Verfügung, um Daten beispielsweise mit einem Funktionsbaustein austauschen zu können. Dazu kann ein Klient eine Anzahl von Variablenobjekten im Server beschreiben, so daß dadurch eine Berechnung gestartet wird,

und nach Beendigung dieser Berechnung die Ergebnisse von anderen Variablenobjekten abholen. Es sind die nachfolgend aufgeführten Services verfügbar (RPC-Prozedurnummer in Klammern):

KS\_EXGDATA (0x00000103)

Dienstgruppe	Server	Manager
Manager	—	ERF

In der Dienstgruppe "Manager" sind alle Dienstleistungen zum Registrieren und Erfragen von KS-Servern zusammengefaßt. Es sind die nachfolgend aufgeführten Services verfügbar (RPC-Prozedurnummer in Klammern):

KS\_REGISTER (0x0000FF01)

KS\_UNREGISTER (0x0000FF02)

KS\_GETSERVER (0x0000FF03)

### 3 Der Objektbaum

Im Folgenden wird der von der KS-Spezifikation vorgeschriebene Teil des Objektbaumes näher erläutert. Für jedes Objekt wird neben dem Objekttyp und dem Namen in eckigen Klammern angegeben, ob das Objekt erforderlich („[ERF]“) oder optional („[OPT]“) ist. Aufgrund der hierarchischen Struktur des Objektbaumes sind dabei folgende Regeln zu beachten:

- Lautet der Implementierungsstatus eines Objekt „erforderlich“ und ist der Status des Elternobjektes „optional“, dann muß das betrachtete Objekt nur dann zwingend implementiert werden, wenn auch das Elternobjekt implementiert wird.
- Lautet der Implementierungsstatus eines Objektes „optional“ und ist der Status des Elternobjektes „optional“ oder „erforderlich“, dann kann (!) das betrachtete Objekt implementiert werden (sobald das Elternobjekt ebenfalls implementiert ist).
- Die beiden vorangegangenen Regeln sind entsprechend rekursiv solange anzuwenden, bis ausgehend vom betrachteten Objekt die Root-Domain erreicht ist.

Die Angabe des Kürzels „[RO]“ bei Variablenobjekten zeigt an, daß nur lesend darauf zugegriffen werden kann, wohingegen bei Angabe von „[RW]“ sowohl ein lesender als auch schreiben-der Zugriff erlaubt ist.

Bei Variablenobjekten vom Typ KS\_VT\_STRING, die mit dem Kürzel „[PA]“ (für „printable ASCII“) gekennzeichnet sind, gelten die folgenden Annahmen: Die im KS-Variablenobjekt abgelegte Zeichenkette darf nur ASCII-Zeichen mit Zeichencodes zwischen 32 (Leerzeichen) und 126 (Tilde) enthalten. Daneben sind noch die Steuerzeichen mit den Codes 13 (CR = „carriage return“), 10 (LF = „line feed“) sowie 9 (HT = „horizontal tabulator“) zulässig. Ein CR oder LF sowie die Steuerzeichenkombinationen CR/LF und LF/CR zeigen den Beginn einer neuen Zeile an. Das Steuerzeichen HT entspricht derjenigen Anzahl von Leerzeichen, die erforderlich sind, um die nächste Tabulatorposition zu erreichen. Alle anderen ASCII-Steuerzeichen sind nicht zulässig.

#### 3.1 Manager und Server

Domain "/" [ERF]

Die Root-Domain muß immer vorhanden sein.

Domain "/acplt" [-]

Diese Domain ist für den Lehrstuhl für Prozeßleittechnik reserviert.

Domain "/vendor" [ERF]

Diese Domain bildet den Ausgangspunkt innerhalb des Objektbaumes, um Informationen

über einen Server oder Manager (Hersteller, Versionsnummer, verfügbare Dienstgruppen, und so weiter...) zur Verfügung zu stellen.

Variable `"/vendor/server_name"` [ERF, RO]

Der Name dieses Servers (Datentyp `KS_VT_STRING`), so wie er sich beim Manager anmeldete. Sollte der Server sich mehrfach beim Manager mit verschiedenen Namen, aber gleicher Portnummer registrieren, so sollte hier der Name der aktuellen Serverversion stehen.

Variable `"/vendor/server_version"` [ERF, RO]

Die Versionskennung des Servers in textueller Form (daher ist die Variable vom Datentyp `KS_VT_STRING`).

Variable `"/vendor/server_description"` [OPT, PA, RO]

Diese Variable vom Typ `KS_VT_STRING` enthält optional eine textuelle Beschreibung dieses Servers.

Variable `"/vendor/name"` [ERF, PA, RO]

Der Name des Herstellers des Servers (Datentyp `KS_VT_STRING`).

Variable `"/vendor/contact"` [OPT, PA, RO]

Wenn vorhanden, dann enthält diese Variable vom Datentyp `KS_VT_STRING` eine Kontaktadresse des Server-Herstellers (beispielsweise eMail-Adresse oder ähnliches).

Variable `"/vendor/disclaimer"` [OPT, PA, RO]

Wenn vorhanden, können Anwender hier noch einmal die genauen Bedingungen des Haftungsausschlusses nachlesen (Datentyp `KS_VT_STRING`). Bitte beachten Sie, daß diese, dem deutschen Recht entsprechend, nicht sittenwidrig sein dürfen.

Variable `"/vendor/server_time"` [OPT, RO]

Wenn vorhanden, dann spiegelt diese Variable vom Typ `KS_VT_TIME` die aktuelle Uhrzeit des Servers wieder.

Variable `"/vendor/av_modules"` [OPT, PA, RO]

Diese Variable vom Typ `KS_VT_STRING_VEC` enthält eine Liste der Kennungen von A/V-Modulen, die der Server unterstützt. Vom Kernprotokoll werden nur die Kennungen "none" sowie "simple" definiert.

Domain `"/vendor/reserved"` [OPT]

Unterhalb dieser Domain können KS-Server eigene, herstellerspezifische Erweiterungen einfügen. Es bleibt der jeweiligen Implementation überlassen, ob sie diesen Zweig benutzt und welche Objekte sie in ihm zur Verfügung stellt.

### 3.2 Protokollerweiterungen

Domain `"/vendor/extensions"` [OPT, RO]

Diese Domain enthält – wenn vorhanden – Informationen über das KS-Kernprotokoll sowie alle Protokollerweiterungen eines Servers. Bietet ein Server Protokollerweiterungen an, so muß auch diese Domain zwingend vorhanden sein. Jede einzelne Protokollerweiterung "xxx" wird – ebenso wie auch das Kernprotokoll – durch eine Domain mit dem Namen "xxx" unterhalb der Domain `"/vendor/extensions"` repräsentiert.

Domain `"/vendor/extensions/xxx"` [OPT, RO]

Jede der Domains, die eine Protokollerweiterung "xxx" repräsentiert, besitzt zwei Kindobjekte, die Einzelheiten des Protokolls beschreiben. Diese beiden Kindobjekte werden im Folgenden näher erläutert:

Variable `"/vendor/extensions/xxx/major_opcode"` [ERF, RO]

Hierbei handelt es sich um ein Variablenobjekt vom Typ `KS_VT_INT`, das den Major Opcode für die Protokollerweiterung mit dem Namen `"xxx"` enthält.

Variable `"/vendor/extensions/xxx/services"` [OPT, RO]

Diese Variable vom Typ `KS_VT_STRING_VEC` enthält eine Liste von Kennungen derjenigen Servicegruppen, die der Server für diese Protokollerweiterung zur Verfügung stellt. Die gültigen Kennungen werden von der jeweiligen Spezifikation der Protokollerweiterung festgelegt. Fehlt dieses Objekt, so sind nur die in der entsprechenden Spezifikation aufgeführten Standard-Servicegruppen verfügbar.

Variable `"/vendor/extensions/ks_core/major_opcode"` [ERF, RO]

Dieses Variablenobjekt vom Typ `KS_VT_INT` enthält den Major Opcode für das KS-Kernprotokoll. Der Wert ist per Definition immer `0x0000`.

Variable `"/vendor/extensions/ks_core/services"` [OPT, RO]

Diese Variable vom Typ `KS_VT_STRING_VEC` enthält eine Liste von Kennungen derjenigen Servicegruppen, die ein KS-Server zur Verfügung stellt. Für eine Auflistung der definierten Servicegruppen und ihrer Kennungen siehe Abschnitt 2.

### 3.3 Managerspezifische Objekte

Die folgenden KS-Objekte sind nur im Objektbaum eines ACPLT/KS-Manager vorhanden. Sie geben Aufschluß über die momentan bei einem Manager registrierten Server.

Domain `"/servers"` [ERF]

Diese Domain fungiert als Sammelstelle für Informationen über alle bei einem Manager zur Zeit registrierten Server. Jeder dieser Server `"xxx"` wird dabei durch eine Domain mit dem Namen `"/servers/xxx"` repräsentiert. Sind beim Manager mehrere Server mit unter gleichen Namen registriert, so werden die einzelnen Server durch ihre Versionsnummern unterschieden.

Domain `"/servers/xxx"` [ERF]

Eine derartige Domain bildet einen Container für Informationen über einen (beziehungsweise mehrere) Server namens `"xxx"`. Eine weitere Unterscheidung wird (soweit erforderlich), über eine Unterdomain erreicht, deren Namen aus der vom Server verwendeten KS-Protokollversionsnummer `"ppp"` gebildet wird.

Domain `"/servers/xxx/ppp"` [ERF]

Jeder Server wird nicht nur über seinen Namen, sondern auch zusätzlich über die Versionsnummer `"ppp"` des verwendeten KS-Kernprotokolls identifiziert. Benutzt ein Server namens `"acme"` beispielsweise die Protokollversion 1, so stehen weitere Informationen über den Server unterhalb der Domain `"/servers/acme/1"` zur Verfügung.

Über jeden Server werden vom Manager die folgenden Informationen im Objektbaum zur Verfügung gestellt:

Variable `"/servers/xxx/ppp/expires_at"` [ERF, RO]

Hier ist der Zeitpunkt abgelegt (Type `KS_VT_TIME`), bis zu dem sich der Server spätestens wieder beim Manager zurückgemeldet haben muß.

Variable `"/servers/xxx/ppp/port"` [ERF, RO]

Diese Variable enthält die Portnummer, unter der der Server erreichbar ist (Variablentyp `KS_VT_INT`).

Variable `"/servers/xxx/ppp/living"` [ERF, RO]

Diese Variable zeigt an, ob der Server noch lebt (1 = okay, 0 = wahrscheinlich tot, Variablentyp `KS_VT_INT`).

## 4 KS-Fehlercodes

Die nachfolgend aufgeführten Fehlercodes sind im Kernprotokoll von ACPLT/KS definiert:

KS\_ERR\_OK = 0x0000

Es ist kein Fehler aufgetreten.

KS\_ERR\_GENERIC = 0x0001

Es ist ein nicht näher spezifizierter Fehler aufgetreten. Als Ursache kommt beispielsweise ein Ressourcenmangel (zuwenig Speicherplatz oder ähnliches) in Frage.

KS\_ERR\_BADAUTH = 0x0002

Der Klient konnte sich nicht ordnungsgemäß authentifizieren. Die Ausführung des angeforderten Dienstes wurde verweigert.

KS\_ERR\_UNKNOWNAUTH = 0x0005

Der Klient benutzt in seiner Serviceanforderung ein A/V-Schema, das vom Server nicht unterstützt wird.

KS\_ERR\_NOTIMPLEMENTED = 0x0003

Der Klient versuchte, einen Dienst aufzurufen, der im Server nicht implementiert ist. Möglicherweise benutzt der Klient auch eine ungültige Dienstenummer.

KS\_ERR\_BADPARAM = 0x0004

Ein oder mehrere der in der Dienstanfrage spezifizierten Parameter ist/sind ungültig. Die zulässigen Werte einzelner Parameter sind bei der jeweiligen Dienstbeschreibung aufgeführt.

KS\_ERR\_BADNAME = 0x0010

Es wurde ein Name angegeben, der ungültige Zeichen enthält. Der Name kann dabei ein Servername oder ein Variablen-/Objektnamen sein. Die jeweils zulässigen Zeichen in diesen Namen sind im Zusammenhang mit dem zugehörigen Dienst dokumentiert.

KS\_ERR\_BADPATH = 0x0011

Der angegebene Suchpfad für den GetOD-Dienst existiert nicht. Enthält der Suchpfad ungültige Zeichen, so kann statt dessen auch der Fehlercode KS\_ERR\_BADNAME auftreten.

KS\_ERR\_BADMASK = 0x0012

Die beim GetOD-Dienst angegebene Maske (regulärer Ausdruck) ist ungültig.

KS\_ERR\_NOACCESS = 0x0013

Mit der aktuellen Authentifizierung ist der gewünschte Zugriff (schreibend oder lesend) auf ein Kommunikationsobjekt nicht erlaubt.

KS\_ERR\_BADTYPE = 0x0014

Der Typ der in ein Variablenobjekt zu schreibenden Daten weicht vom Typ des Variablenobjektes ab und der Server kann keine Konversion zwischen den beiden Datentypen ausführen. So kann in ein Variablenobjekt, das eine Ganzzahl vom Typ KS\_VT\_INT aufnimmt, nicht mit einem Vektor von Ganzzahlen (Typ: KS\_VT\_INT\_VEC) der Länge Eins beschrieben werden.

KS\_ERR\_CANTSYNC = 0x0015

Beim EXGDATA-Dienst wurden Variablenlisten angegeben, deren Variablen zu verschiedenen Funktionsbausteinen oder dergleichen gehören und der Server kann die EXGDATA -Dienstanforderung in dieser Variablenkonstellation nicht ausführen.

KS\_ERR\_NOREMOTE = 0x0020

Es ist unzulässig, sich als Server bei einem Manager anzumelden, der auf einer anderen Maschine läuft.

KS\_ERR\_SERVERUNKNOWN = 0x0021

Es ist kein Server mit den gesuchten Eigenschaften beim Manager registriert.



## 5 AuA – Abkürzungen und Akronyme

ACPLT/KS	Kommunikationssystem des Lehrstuhls für Prozeßleittechnik der RWTH Aachen
IP	Internet Protocol
ISO	International Standards Organization
ONC	Open Network Computing
OSI	Open Systems Interconnect
RPC	Remote Procedure Call/Calling
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XDR	External Data Representation

## 6 Literaturverzeichnis

- [1] Bloomer, John:  
Power Programming with RPC.  
O'Reilly & Associates, Inc., Sebastopol, CA, 1992