

해결안

- Long data type
Tuxedo가 32-bit인 경우는 Jolt의 integer data type을 사용
Tuxedo가 64-bit인 경우는 Jolt의 double data type을 사용
- char data type 처리
Java "char" primitive type은 16-bit Unicode 값이나 C/C++는 8-bit ascii 기반의 값이다.
이러한 차이 때문에 Java에서 "char"가 아닌 "byte"를 사용해야 한다. "char" 를 위해서는 다음 method를
addByte, setByte, setByteItem, getByteDef, getByteItemDef

대량 데이터 전송 방법



제품	TOP END 2.05/3.0
사용 OS	모든 플랫폼(platform)
작성일	2003/01/20
이슈사항	TOP END에서는 대량 데이터 전송을 위해 LMA를 지원하며 이 기능을 사용하는 경우 2GB까지 전송이 가능하다.

해결안

1. LMA functionality를 enable 시킴

- * nm_config
 - [Imw timeout]
노드간 LMA control message를 받거나 attachment read/write operation을 complete timeout이 되면 failure recovery operation을 수행, default는 300초
 - [Imm port]
Large Message Manager(LMM)의 port number
 - [Imm address]
5개까지 기술할수 있음. 첫번째가 실패했을때만 두번째 세번째...
 - 예
...
[Imw timeout] 300
[Imm port] 5000
[Imm address] 123.456.789.123

2. 프로그램

- LMA를 사용하기 위해서는 API Version 2를 사용해야 한다. 아래와 같이 "#include <tp_csi.h>위에 선언
#define TP_API_VERSION 2
#include <tp_csi.h>
- attach_info
tp_attach_type : file, pipe, etc
tp_attach_size : byte
0은 전체

해결안

- info->tp_flags에 TP_ATTACHMET가 되어 있는지 아래와 같이 check하도록 한다.
if(dif_struct->info->tp_flags & TP_ATTACHMENT) {
TP_ATTACHMET가 설정되어 있는데 attachment를 처리하지 않으면 TP_PROTOERR 발생
- attach_info->tp_attach_id는 수정할 필요가 없으나 새로운 attachment인 경우 아래와 같이 "0"으로 채운다.
memset(&dif_struct->attach_info->tp_attach_id, '0', sizeof(tp_attach_id_t));
- Windows에서는 PATH 기술 시 아래와 같이 "W"는 "WWW"으로 기술
strcpy(dif_struct->attach_info->tp_attach_location.tp_attach_path,
"d:WWWappsWWWbeaWWWtopendWWWimaWWWsdata.txt");
- 예
memset(&dif_struct->attach_info->tp_attach_id, '\0', sizeof(tp_attach_id_t));
dif_struct->attach_info->tp_attach_type = TP_ATTACH_TYPE_FILE;
dif_struct->attach_info->tp_attach_size = 0;
strcpy(dif_struct->attach_info->tp_attach_location.tp_attach_path,
"d:WWWappsWWWbeaWWWtopendWWWimaWWWsdata.txt");

추가정보

데이터 전송 시 별개의 session과 process에 의해 진행이 됨

MSR 사용 방법



제품	TOP END 2.05/3.0
사용 OS	모든 플랫폼(platform)
작성일	2003/01/20
이슈사항	TOP END에서는 message 값에 의한 routing을 위해 MSR을 지원하며 크게 두 가지 형태로 사용할 수 있다. 1. application에서 target name을 직접 기술하거나 2. tp_services 파일을 이용하는 방법이다.

해결안

1. MSR할 AC의 configuration file에 있는 [target]에 target name을 기술

```
ac_config
...
[automatic throughput optimization] W
off 1 10 1 5 120
[target name] TargetA
```

2.1 프로그램에서 target을 결정하는 경우는 tp_message_routing_qualifer에 target name 기술(이 경우는 여기까지만 하면 됨)

```
...
space_fill(dif->service->tp_message_routing_qualifier, "TargetA", 8);
...
```

2.2 미리 지정된 것을 사용하는 경우(tp_services 파일)는

1) target name을 space로 초기화

```
...
space_fill(dif->service->tp_message_routing_qualifier, " ", 8);
...
```

2) tp_services 파일 작성(자세한 내용은 매뉴얼 참조)

```
<<services section>>
product "banksys" function "TOUPPER" use method "CASE1";
```