APPC Interface

The APPC Interface is described under the following headings:

- Concepts
- Implementation
- Requirements

Concepts

APPC is a communication protocol specifically designed for distributed processing. While 3270-based communications are hierarchical (controlled by the host), APPC is peer-to-peer, meaning that programs communicate as equals. APPC-based applications send and receive data directly, without using 3270 screens. In APPC, the communication between two programs is called a conversation. The program that initiates the conversation is called the client, and the program that responds to the client is called the server. A distributed application is an application that requires programming from different places. A transaction program (TP) is one of the programs in a distributed application. A transaction is a business deal cooperatively completed by two or more transaction programs.

An APPC application can be designed to provide the functions of a 3270-based application, but without the same performance and reliability problems associated with 3270 communications. APPC programs can communicate with each other no matter which API is used by either side.

APPC data transfers are not restricted by screen size. Applications can send up to 32 kbytes of data in a single send. This alone greatly increases the performance of APPC over 3270. Additionally, APPC automatically buffers the data that is sent to the partner program. Buffering is performed to optimize network data flow for client/server applications. If a number of records are sent using several send calls, the data can be sent as a single network flow.

Because APPC is a peer-to-peer protocol, both APPC applications must agree on which communication flows will be used for the conversation. The communications flow includes who will start the conversation, what each side will say, when and how they will say it, and who will end the conversation. With APPC, any supported platform can have applications that function as either the client or the server.

APPC does not assume that applications will provide data in a specific format, since APPC can transport data in any format. When communicating between platforms, APPC does not perform data conversions. Although data conversion routines are simple to code, the fact that conversion may be needed is often overlooked when first developing APPC applications. The developer needs to include the ability to convert character, integer and floating point data when sending data between platforms.

Similarly, binary data must be handled differently in APPC. 3270 applications do not have to deal with binary data, since all data is expanded to text format for the 3270 screens. When migrating a 3270 application to APPC, it is often necessary to expand the binary data to text before sending it to the partner platform. Since different platforms store binary data in different formats, sending binary data in its native format across platforms can cause application errors.

There are several ways to ensure that both partners can correctly handle transported data:

- An organization can define a standard for all data that is exchanged, so an application can convert between the standard and the format for the platform on which the application is running.
- All numbers can be sent in character form rather than binary.
- Use system and language tools that help translate data to be sent to other machines.

Implementation

Com-pletes APPC Interface allows online programs to act as servers in an APPC conversation. In general the server program running under Com-plete does not know or does not need to know wether it is communicating with a terminal or an APPC partner. The data transformation is done later in the corresponding device modules according to TIB specifications. Existing 3270 application programs might work correctly as APPC server TPs. Clients must of course be to understand the existing logic of data flow. No 3270 control data is sent unless the TP inserts it documentationly (Write Special).

Requirements

To enable the Com-plete APPC Interface , the APPC=YES parameter must be present on the APPL definition of Com-plete and LOGMODE entries for SNASVCMG and a user LOGMODE must have been included in an active logmode table.

A buffer pool with an element size of 32K, location ANY must be specified in SYSPARMS for the APPC Receive Buffers.

The TP name length for servers running under Com-plete is restricted to 8 characters. TP names longer than 8 characters are truncated to 8.

ULOG is invoked prior to attaching the requested TP. A valid userID/PASSWORD combination must be present in the security fields of the ATTACH request. If logon fails the ATTACH is rejected with the corresponding sense code. On normal or abnormal termination of the requested TP the standard logoff procedures are invoked before the conversation is deallocated.

Com-plete fully supports BASIC and MAPPED conversations with Synclevel=None. Synclevel=CONFIRM is currently supported by the interface but API support is not yet available. CONFIRMD requests are generated automatically by the interface when appropriate.

Application Data GDS(12FF) is expected after the ATTACH FMH-5. If a User Control Data GDS (12F2) is found Com-plete assumes this is a CICS Transaction Routing Request and activates the supplied transaction URTE. UserID/PASSWORD in the security fields are then ignored. They will be taken from CICS input data.