ABAP for Functional Consultants

Anthony Cecchini, President, Information Technology Partners

Founded in 1993, Women Owned 8(M), Small Business Certified with a GSA IT 70 Schedule, we focus solely on SAP. IT Partners has 20 years of experience helping both large and midsize companies, and Government Entities.

Anthony has 17 years of experience in SAP R/3 business process analysis and SAP systems integration. His areas of expertise include SAP NetWeaver integration; ALE development; RFC, BAPI, IDoc, Dialog, and Web Dynpro development; and customized Workflow development.
SAP consultants come in all shapes and sizes from the extreme techie guys that can only communicate in binary code, to those smooth talking ‘gift of the gab’ salesman like functional consultants who run in fear at the thought of discussing anything technical. When a project gets on the way, somehow the two must meet and work together. Being a “Techie” - I want to help bridge this chasm and help functional consultants “Crack the Code” and gain some footing with this presentation.
A functional consultant evaluates the demands in talking with the customer's representatives, transforms the essence into an abstract and algorithmic business model. Hence, he/she identifies the use cases and transforms them into logical views.

Then the main task starts: customizing the respective business area and making sure the system reacts in the manner according to the constraints of the requested use case.

Along with this an SAP functional consultant needs to develop functional specifications.
Technical consultants plan the technical requirements for an object with the Functional Consultant and the manager of the technical team and then carry out the required technical tasks in the system. Depending on the scope and complexity of the implementation, technical consultants may work in several areas, for example, system administration, database administration, network administration, operating system administration, development of cross-application components, or ABAP Development.

Along with this an SAP Technical consultant needs to develop Technical specifications.
While the Functional Consultant defines and isolates the GAP in the Use Case, the Technical Consultant decides the best Technical Realization in achieving the “Goal” or end state desired.

My assertion is, the quality, stability, and overall integrity of the process is increased (x) fold by collaboration from start to finish.
What will we discuss?

ABAP
- Client Server Architecture
- ABAP Repository
- ABAP Workbench
- Introduction to ABAP Programming Language

Data Dictionary
- Structures
- Internal Tables
- Transparent Tables

User Exits and Badi's

ALE & IDocs

ABAP Debugger
The **Presentation layer** contains user interface like SAP GUI, and the Enterprise Portal for the user to view the data from the application layer.

The ABAP programs run in the **Application layer**. The ABAP programs then read the data needed for processing from the database.

**Database layer** is the lowest level. Data is managed using the relational database management system (RDBMS) concept.
The Repository consists of all system development objects in the SAP R/3.

Some development objects are:
- Report Program
- Function Groups and Function Modules
- DDIC Tables
- Dynpro Components
- Classes & Methods
- Packages, etc.

The repository is divided according to the application components. Within the application component there are several packages which contain relevant object for logical subdivision. Whenever an application object is developed it must be assigned to a package.
Searching the Repository

There is a Search tool you can use to search the repository information system.

This is used to make random searches like “search for all programs by a particular package ...”

To get to the repository information system, follow the menu path from SAP access Menu. Tools->ABAP Workbench->Overview->Information system or use transaction SE84 or SE80
ABAP Workbench

ABAP Workbench includes tools that are required for the creation and editing of repository objects.

These tools cover the entire software development life cycle.

**ABAP Editor:** It is used for editing the source code.

**ABAP Dictionary:** It is used for creation of database table definitions, data types, data elements and so on.

**Screen painter:** It is used for configuring screen along with the functions for user dialogs.

**Menu painter:** It is used for defining user interfaces like menu bar, standard toolbar, application tool bar and function key settings.

**Function Builder:** It is used for creation and maintenance of function modules.

**Class builder:** It is used for maintaining global classes and interfaces.
ABAP Workbench

Each of the repository objects can be edited using a respective tool. For our convenience we can access all these tools in The Object navigator, Transaction SE80.

The Object navigator is split into two areas:
Tool Area
Navigation Area
ABAP Workbench

**Navigation Area:** Navigation area on the left hand side of the screen displays the hierarchy of the objects. We can either display or hide the navigation area in the screen by selecting full screen mode on or off. Double clicking on an object in the navigation area displays the object in its corresponding editor in the tool area. For example if you choose a program it will be displayed in the ABAP Editor in the tool area and if you choose a class it will be displayed in the class builder.

**Tool Area:** It is the area for displaying and editing the development object using the appropriate tool like class builder or ABAP editor etc.
Introduction to ABAP Programming Language

REPORT TEST.

WRITE 'Hello World'.

ABAP is platform independent. It is independent of the database and operating system.

ABAP programs have individual statements.

The first word is the ABAP Keyword.

End statements with a period.
Introduction to ABAP Programming Language

Chained statements

Consecutive statements with an identical first (leftmost) part can be combined into a "chained" statement using the chain operator ":=" (colon).

The common part of the statements is written to the left of the colon, the differing parts are written to the right of the colon and separated by commas.

```plaintext
WRITE FLIGHTINFO-CITYFROM.
WRITE FLIGHTINFO-CITYTO.
WRITE FLIGHTINFO-AIRPTO.

WRITE: FLIGHTINFO-CITYFROM,
       FLIGHTINFO-CITYTO,
       FLIGHTINFO-AIRPTO.
```
Introduction to ABAP Programming Language

Where do you think ABAP programs Run?
Introduction to ABAP Programming Language

Where do you think ABAP programs Run?

Where is the Data the ABAP program needs?
Introduction to ABAP Programming Language

Where do you think ABAP programs Run?

Where is the Data the ABAP program needs?
ABAP is *platform independent*. It is independent of the database and operating system...

How do we read and write data from the RDBMS?
OPEN SQL

Open SQL consists of a set of ABAP statements that perform operations on the central database in the SAP System.

The results of the operations and any error messages are independent of the database system in use.

** Program: BOOKINGS
** Author: Joe Byte, 07-Jul-2007

REPORT BOOKINGS.

* Read flight bookings from the database

SELECT * FROM FLIGHTINFO
WHERE CLASS = 'Y' "Y = economy" OR CLASS = 'C' "C = business"

(…)

******************************************************************************************
** Program: BOOKINGS**
** Author: Joe Byte, 07-Jul-2007**
******************************************************************************************
Final Thoughts..
ABAP does not differentiate between upper and lowercase for keywords.

Comment lines begin with a ‘*’

Double quotes indicate the rest of the statement is comment

REPORT bookings.

* Read flight bookings from the database
SELECT * FROM FLIGHTINFO
WHERE CLASS = 'Y' "Y = economy"
  OR CLASS = 'C'. "C = business

(...)
## Introduction to ABAP Programming Language

### Bonus Slide
Commonly used **System Fields** in ABAP

<table>
<thead>
<tr>
<th>System field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sy-subrc</td>
<td>After <em>every</em> Open SQL statement, the system field SY-SUBRC contains the value 0 if the operation was successful.</td>
</tr>
<tr>
<td>sy-tcode</td>
<td>Current transaction</td>
</tr>
<tr>
<td>sy-datum</td>
<td>Local date of system</td>
</tr>
<tr>
<td>sy-usize</td>
<td>Local time in system</td>
</tr>
<tr>
<td>sy-repid</td>
<td>Name of the current program</td>
</tr>
<tr>
<td>sy-langu</td>
<td>Logon language</td>
</tr>
<tr>
<td>sy-mandt</td>
<td>Logon client</td>
</tr>
<tr>
<td>sy-uname</td>
<td>Logon Name of user</td>
</tr>
</tbody>
</table>
What will we discuss?

ABAP
- Client Server Architecture
- ABAP Repository
- ABAP Workbench
- Introduction to ABAP Programming Language

Data Dictionary
- Structures
- Internal Tables
- Transparent Tables

User Exits and Badi's

ALE & IDocs

ABAP Debugger
What is a Data Dictionary?

A data dictionary in computing terms is the source of information in which the system data is defined in a logical way. The data dictionary is the core of a well-structured development environment. The elements that make up a dictionary are known as metadata.

The ABAP data dictionary data is the core of the R/3 development system. The integration of the ABAP dictionary with the development and runtime environments is an active integration.
Structures

Structures allows to combine values that logically belong to one data object.

You can define a structure using transaction SE11.

When defined using SE11, it becomes part of the Repository and can be used by ALL developers. These are referred to as “GLOBAL”
Structures

Structure variables are defined in the ABAP program with DATA statements

DATA wa_zaddress TYPE zaddress.  Using Global Type

“LOCAL” Structure variables are defined in the ABAP program with DATA statements

DATA BEGIN OF ls_zaddress,
    name   TYPE surname,
    street(30) TYPE c,
    city   TYPE spfli_type-cityfrom,
END OF ls_zaddress.
Internal Tables

Internal table is a data object to keep identically structured data records at runtime.

ABAP internal tables are **STORED IN MEMORY** not on the RDBMS.

Individual records are known as table rows or table entries.

Used in retaining data from database tables or sequential files for future processing.

```plaintext
DATA gt_tab  TYPE  <Type> OCCURS 0 TIMES.

DATA gt_itab  TYPE  TABLE OF <Table_type>.

DATA gt_itab  TYPE  TABLE OF <Struc_type>.
```
Internal Tables

ABAP internal tables are **STORED IN MEMORY** not on the RDBMS.
Internal Tables

You can ADD lines to an Internal Table using the APPEND or INSERT Commands.
**Internal Tables**

You can READ, CHANGE, or DELETE rows of an Internal Table.

You also serially read an Internal Table using the LOOP AT – ENDLOOP Construct.

```
LOOP AT <ITAB> INTO <WA>
  WHERE <Condition>.
  *Process the Data From <WA>
  ENDLOOP.
```
ABAP internal tables are **STORED IN MEMORY** not on the RDBMS.
Transparent Tables

Transparent tables defined using transaction SE11

Transparent tables defined in the application as a part of the corresponding database table that stores the actual data.

The fields of the transparent table point to the corresponding columns on the database table.

Data elements refer to domains for technical properties.
Transparent Tables

In addition to list of fields we need more information to create a database table some of them are –

- Key fields are to declared at the beginning of the table. This determines the Primary key of the table.
- Technical properties that are needed to create the database table like size and mode of access.
- Secondary indexes and buffering to speed up access.
Transparent Tables

ABAP Program

DATA gs_vbap TYPE vbap.
SELECT ... FROM VBAP
  INTO gs_vbap ....

Transparent table

RDBMS

VBAP

BA 17
SAP HANA

INTRODUCING THE SAP HIGH-PERFORMANCE ANALYTIC APPLIANCE

SAP HANA is the next generation of SAP’s in-memory computing technology. SAP HANA is a multi-purpose, data-source agnostic, in-memory appliance that combines SAP software components optimized on hardware provided, and delivered, by SAP’s leading hardware partners.

Analyze massive amounts of data

3600x faster*
What will we discuss?

ABAP
- Client Server Architecture
- ABAP Repository
- ABAP Workbench
- Introduction to ABAP Programming Language

Data Dictionary
- Structures
- Internal Tables
- Transparent Tables

User Exits and Badi's

ALE & IDocs

ABAP Debugger
USER Exits / BADIs

In SAP, a user exit is a place in a software program where a customer can arrange for their own tailor-made solution to be called.

- USER EXIT – Form or function call
- BADI - Business add-ins are enhancements to the standard version of the system using OOPs Concept
USER Exits / BADIs

The R/3 enhancement concept allows you to add your own functionality to SAP’s standard business applications without having to modify the original applications.

Advantage -

- They do not affect standard SAP source code
- They do not affect software updates and upgrades
What will we discuss?

ABAP
- Client Server Architecture
- ABAP Repository
- ABAP Workbench
- Introduction to ABAP Programming Language

Data Dictionary
- Structures
- Internal Tables
- Transparent Tables

User Exits and Badi's

ALE & IDocs

ABAP Debugger
**ALE and IDocs**

Application Link Enabling Enables SAP to communicate to other systems.

Supports data consistency and data availability in distributed systems.

Integrating system through asynchronous messaging using IDOCs. (EDI)

Enables distribution between systems with different versions.

Provides functions for administration, monitoring and development.

Library of Delivered ALE business processes to cover important business functions.
ALE and IDocs

An Intermediate Document is a container for the data of a business object or technical R/3 object.

Each IDOC has a message type. This indicates the type of business object or the business function of the data.

Message types have processing rules in the receiving system.

An IDOC contains segment hierarchy. The IDOC type describes the technical structure of the IDOC.

IDOC types have versions.
ALE and IDocs

An IDoc consists of 3 types of records.
Control Record.

- One Control Record per IDoc.
- Data Record. Multiple data records per IDoc.
- Data Records contain Data to be transferred
- Status Record. Status Record will have give the status of the IDoc.

IDoc status can be checked using TCodes WE02, WE05 and WE09

IDocs can be reprocessed using TCode BD87
**ALE and IDocs**

Common IDoc Status

- **03** - Data passed to port OK
- **12** - Dispatch OK
- **51** - Application document not posted
- **68** - Error - no further processing
- **70** - Original of an IDoc which was edited

You can get a complete list and details by using Transaction Code WE47.

Transaction WE02 0r WE05 (IDoc Display)
What will we discuss?

**ABAP**
- Client Server Architecture
- ABAP Repository
- ABAP Workbench
- Introduction to ABAP Programming Language

**Data Dictionary**
- Structures
- Internal Tables
- Transparent Tables

**User Exits and Badi's**

**ALE & IDocs**

**ABAP Debugger**
‘Debugging’ is a term, which is directly associated with programming. It is a general term used by all Software Professionals, irrespective of the language and platform. ‘Debugging’, as the name suggests, is used to debug the program.

In other words it is slow motion, step by step execution of the program, so that one can closely find, how the program/software is working. Debugging is typically used to fix the Bugs in the program but in SAP debugging is invariably used for understanding the flow of program itself.

In SAP, debugging had been a quite easy task. In SAP there are various ways to debug a program.
ABAP Debugger
Debugger in SAP can be started by typing ‘/h’ in the command field before execution.
Initially, Dynpro Flow logic is displayed in the window if debugging has been started for module pool. If a report program is being debugged then direct program code is displayed.
Before going into details, it is necessary to discuss the term ‘BREAK-POINT’.

A break point is a technique to stop the execution at particular line in the code during the debugging.

Just double-click on the line of code you want the debugger to stop at when the transaction is executed.
ABAP Debugger

So now I have executed ME53 again and the debugger has stopped at the line of code where I placed a BREAK-POINT.

Now I can double-click on EBAN-BANFN and I can see it’s value.

This can be done for Internal Tables, Structures and Transparent Tables.
There are four different ways in which you can go through the source code of a program you want to analyze:

<table>
<thead>
<tr>
<th>Single step</th>
<th>Use this option to execute the program statement by statement. This allows you to branch to other program units.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute</td>
<td>Use this option to process a program line by line. All of the statements on the current line are processed in a single step. This allows you to process the whole program.</td>
</tr>
<tr>
<td>Return</td>
<td>The Debugger returns to the point at which control is passed back to the main program. Use this option to return from other program units.</td>
</tr>
<tr>
<td>Continue</td>
<td>Use this option to process the program up to the next breakpoint. If there are no more breakpoints in the program, the system exits debugging mode and executes the rest of the program normally.</td>
</tr>
</tbody>
</table>
ABAP Debugger

Debugging can be done for programs scheduled to run in background.

Job should be either in Scheduled or Released status.

After scheduling the program to run in background, select the job in SM37 and enter jdbg in the command area.

Now the debugger is activated.
**ABAP Debugger**

Running programs (i.e. jobs with status Active) can also be debugged.

Use Transaction SM50 to get the list of processes currently running.

Identify the process related to the program you want to debug and click or highlight.

Use Menu Path: **Administration ➔ Program ➔ Debugging**

The Debugger will be activated.
What will we discuss?

**ABAP**
- Client Server Architecture
- ABAP Repository
- ABAP Workbench
- Introduction to ABAP Programming Language

**Data Dictionary**
- Structures
- Internal Tables
- Transparent Tables

**User Exits and Badi's**

**ALE & IDocs**

**ABAP Debugger**
Why bother to learn this?

Let me highlight some of the benefits that learning some of the techie stuff will do for you, your colleagues and customers.

- First of all, learning a new skill can only be positive for your career. If you want to move into a new role within your company or move onto another organization, adding to your skill set is always a great thing to do.

- When your customers want something new or change and existing process, you will be able to give a much more informed response and be able to challenge them from a slightly different angle.

- The techie guys will be more willing to discuss other possible solutions with you as they will see that you understand what they are saying. This will quite often result in a better solution that you can offer your customers.
Let me highlight some of the benefits that learning some of the techie stuff will do for you, your colleagues and customers.

- We all know that sometimes the work estimates given for some development work can be exaggerated somewhat. Don’t let this happen at your expense; challenge the estimates by making it known you know what you are talking about and that you have a good idea just how much work should be involved for certain types of development work.

- Quite often, your customers invite their own techie guys to meetings. Being able to express possible solutions in a technical manner can help build their confidence in you and your company.

- By learning and having a good grasp of the underlying SAP technologies that are used throughout the SAP landscape, you will be in a much better position to understand and get up to speed on new solutions SAP introduce further down the line.

Why bother to learn this?
Thank You & Company Info.

- In Business for 20 years
- Women Owned (8m)
- Small Business Certified
- GSA IT 70 Schedule.
- WWW.ITPSAP.COM

Connect with me on LinkedIn - http://www.linkedin.com/in/ajcecchini
“LIKE US” on Facebook - http://www.facebook.com/itpsapinc
Follow us on Twitter - https://twitter.com/itpsapinc

View my Social Media Press Kit - http://whywebpr.com/anthonycecchini
Thank You & Newsletter.

SAP Challenges Oracle - Is HANA the Secret Weapon?

Take Note (Insights into SAP solutions and Emerging Technology)

SAP AG (SAP) revealed the biggest overhaul to its mainframe enterprise software in two decades, as it tries to remain on its dominance in that market, while in parallel, spawning an attack on Oracle Corp.'s database business.

At an event in Palo Alto, California, SAP unveiled a much better version of its Business Suite software running on top of its HANA database. By landing up business analysts from pricing online tickets to predicting weather, SAP is pushing to replace database software from Oracle, Microsoft Corp. and International Business Machines Corp. (IBM) that customers use to process and analyze information.

The move bolsters the core product, which still accounts for the bulk of SAP's revenue, after a period of focus on the past three years to expand into mobile and Internet-based software services. The company also introduced new apps for its Cloud platform and HANA, in some selling the new features to entice customers to shift to its new repository product.

Strategic Position
Thank You!

- Email us Questions
- Post questions on our FB Page
- Use Twitter
- Use the “TEXT ME” feature in our Media Profile
Are You Smarter Than a ABAPer?
Question

What are the different layers in a 3 tier client server architecture?
Answer

Presentation Layer
Application Layer
Database Layer
In SAP, a ________ is a place in a software program where a customer can arrange for their own tailor-made solution to be called.
In SAP, a **User Exit** is a place in a software program where a customer can arrange for their own tailor-made solution to be called.
Question

TRUE or FALSE

ABAP Transparent tables are STORED IN MEMORY.
Answer

False!
ABAP Internal tables are STORED IN MEMORY.
True or False

Application Link Enabling Enables SAP to communicate to other systems.
True!

It provides this capability by integrating system through asynchronous messaging using IDOCs. (EDI)
Question

How do you start the debugger for a transaction?
Answer

By typing /h in the command line

By setting a BREAK-POINT
Thanks for Playing!