What is ABAP?

Originally: Allgemeiner Berichts-Aufbereitungs-Prozessor (Generic Report Preparation/Generation Processor)

Now: Advanced Business Application Programming

Created by SAP initially for report creation. Later evolved into full language. Made public for program customization and enhancement by companies.

Notable for its database abstraction (logical databases) and ease of database interaction. Embedded SQL statements in code (Open SQL). Platform independent.

Extended in 1998 to include Object support (ABAP Objects).

Developing using OO model recommended, but legacy procedural coding still supported.
Learning ABAP—NetWeaver Programming Core

NetWeaver Training Competencies and Options
What is SAP NetWeaver?

Technical foundation for SAP applications (ERP, CRM, SCM, etc.).

- Provides internal connectivity (integration) between SAP modules and external connectivity with other systems.
- Eases integration between systems.
- Provides framework for development.
- Supports development in Java and ABAP.
SAP Basis Admin—sample job ad

SAP Basis Administrator, Junior
Linux, OS Knowledge, Database/Oracle Knowledge

Position Overview
The responsibility of this position is for implementing, documenting, maintaining, and supporting the technical environment required by SAP. Individuals in this position work at a technical level with an understanding of SAP, its underlying database, operating system, and hardware platform. Incumbents interact with the SAP development teams, SAP configuration teams, various technical teams, and business stakeholders to optimize and maintain the overall SAP reliability, availability, and performance.

Requirements:
As a positive, self-motivated member of our support team, you will learn all aspects of SAP Basis administration while supporting, monitoring, and administering our install base of customers.

- Minimum of three (3) years IT experience or Bachelor’s degree in IT/MIS
- Work collaboratively across project teams
- Plan and organize work & deliverables
- Solve problems through research and technical detective work
- Ability to learn new technologies and processes quickly
- Design integration scenarios using the appropriate integration tools and protocols
- SAP Basis, DBA, or extensive Operating System experience highly desired.

3-tiered architecture

Presentation Layer

Application Layer

Database Layer
3-tiered architecture + ABAP

Presentation Layer

ABAP Programming & Output

Application Layer

ABAP Execution

Database Layer

ABAP Program & Data Storage

3-tiered architecture + ABAP (General)

Presentation Layer
Run Transaction/Program

Application Layer

ABAP Program
Local Data Variables
Program Logic

Database Layer
Program Repository
Database Table(s)

Based on content from Unit 2, BC100, figures 25-35
Developing in an Established Environment

Programming in an environment with many pre-existing named components.

Name collision can break existing functionality.

Program namespaces beginning A through X are reserved for use by SAP. Y and Z can be used by customers.

Modification/programming only possible with a developer key and appropriate configuration. (SCC4)

Key is client and username dependent.

Keys provided by SAP based on license agreement. May require additional payment.

Typical Company SAP Landscape

Typical landscape involves multiple systems—development (DEV), testing (QAS), production (PROD), training (TRN).

Developed components move from system to system via transports.

Development done within packages which are tied to transport requests. Requests are exported/imported for transportation. (e.g. DEV → QAS)

Transport management key system administration task to ensure dependencies and system integrity maintained.
Typical Company Installation Landscape

Programming Process

1. Create new or select existing package.
2. Create new or select existing transport request for package.
3. Begin creating a new program.
4. Enter developer key to authorize development (if not done previously).
5. Create, debug program. Save.
6. Test. Revise as necessary.
7. Activate program.
8. Transport package.

Items that will not be transported can be assigned to the local package $TMP.

Within our development landscape, transports are essentially to ‘dummy systems’.
Beginning Programming

ABAP Development Rules:
- ABAP programs made up of individual sentences/statements.
- Each statement begins with a keyword.
- Conventional to type keywords in all uppercase letters and variables, operators, etc. in all lowercase.
  
  Built in Pretty Printer
- Each statement ends with a period.
- Statements can be indented as desired.
- Statements can extend over multiple lines.
- Lines with * in column 1 are comment lines.
- Double quotes ("" ) are to-end-of-line comments.

Source: SAP BC100, Page 60-61

ABAP Editor

Various settings can be maintained through Utilities → Settings, ABAP Editor

Launch directly with transaction SE38 or SE80 for ABAP Workbench.

To avoid name collision, in all development done on our system begin your programs with the prefix Zxx_ where xx is your individual user number (the last 2 digits of your user name).

If you do your own practice development, enter source code from various sources, etc. you must comply with this requirement.

If you have access to multiple SAP systems note that this number will vary among systems.

If you want to create multiple instances of the same program (practice, redo, etc.), end instances with _01, _02, etc.
Programming Basics—Begin Creating New Program

**Option 1:** Use transaction SE80 to launch ABAP Development Workbench.

Select Package, and enter the name of the package you wish to work with.

If package not previously created, will be prompted to create

For class: **Zxx_PKGcc** (where c is sequential number starting 00)

Programming Basics—Package and Transport Creation

If no transport request exists, create one. Use search feature if one exists.

Use your name in the Description. The system will provide a request number.
Programming Basics—Program Creation

Right click on package in the Object Name list and select Create → Program.

Program name in ALL CAPS (remember naming rules).
Unless needed (to be discussed later) TOP INCLUDE should be off.

Programming Basics—Creation Sequence

**Title**—brief program description.
**Type**—Executable program
**Status**—Test Program

Save/check this dialog and the one that shows the package and the transport request.
Programming Basics—Begin Creating New Program

**Option 2:** Use transaction SE38 to launch ABAP Editor.

Specify Program Name. Select Create.

![ABAP Editor: Initial Screen](image)

Cannot create a new package with this sequence! But if one already exists, can use it.

Programming Basics—Enter, Check, Pretty Print, Run

Use editor to enter code.

Many right-click options.

Check operation validates syntax.

Pretty Printer ‘cleans up’ code layout.

Direct Processing to run.
Additional ABAP Workbench Notes

Appendix

The following slides show screen shots for reference which are displayed live in lecture.
SE16 showing table TSTCT limited to English

SCC4

Change View "Clients": Details

Client: [Name]
City: [Location]
Logical system: [System]
Client role: [Role]

Changes and Transports for Client-specific Objects
- Changes without automatic recording
- Automatic recording of changes
- No changes allowed
- Changes with automatic recording, no transports allowed

CROSS-CLIENT OBJECT CHANGES
- No changes to cross-client Customizing objects

Protection Client Customization tool
- Protection level: 3

CATT and CATT Revisions
- CATT and CATT Not Allowed

Restrictions
- Locked due to client copy
- Protection against SAP upgrade
SE38 ABAP Editor

SE80 ABAP Workbench