

PRESENTATION #: 5115

USING SAP ON-LINE DOCUMENTATION

Presented by James Lancello

Deloitte & Touche Consulting Group/ ICS
Destec Tower, Suite 700
2500 City West Boulevard
Houston, Tx 77042
(800)658-4273

OBJECTIVES OF THIS PAPER

- **The Current State of Workplace Documentation**
- **The Benefits of SAP R/3 On-Line Documentation**
- **How and where it can be used**
- **Change Requests**
- **Program Documentation**
- **Transport Requests**
- **Closing Remarks**

THE CURRENT STATE OF WORKPLACE DOCUMENTATION

If one were to make an observation regarding documentation in the workplace, it would probably be to suggest that it has the lowest priority in the scheme of things. If performed at all, it is generally completed at the last minute, usually under rushed conditions, lacks the attention to it that was given to the activity it represents, is short on content, not thorough, nor complete and is not easily available to those who need it.

In addition, it is usually stored in hard copy format which ends up cluttering desks, file cabinets, and storage areas. If needed, it requires someone to remember who has it, where they last saw it, how they can get it, if its information is current and accurate, and how it can become available to those who require it. Or, multiple copies are being maintained thereby incurring additional costs to a business, and then it can be found that usually they are not in sync as time progresses.

These are just some of the conditions that prevail out in the workplace but we will not dwell on them. Suffice it to say this, in today's complex work place where it is needed more now than ever, it is costing businesses a great deal of money in abstract ways that are not being recognized. Productivity, paper purchasing, transportation, distribution, usage, recycling and payment, printer wear and tear, etc., are just some of the ways it manifests itself.

At this point, I would like to introduce a method of reducing or eliminating they situations described above. Using SAP R/3 On-Line documentation in place of the current documentation methods when possible.

THE BENEFITS OF SAP R/3 ON-LINE DOCUMENTATION

SAP R/3 On-Line Documentation comes as part of the delivered system. Within it you can enter information specific to your particular environment and make it available to those who require this information. The information you provide is centralized. It is created and maintained in one area. If a hardcopy is desired, it can be printed at any time to any network printer. Formats can be created in SAP and downloaded to your personal computer. There they can be worked on and then uploaded to

SAP once completed. These documents will then become integrated into your information system and immediately available to those who require it.

WHERE AND HOW CAN IT BE USED

If you've ever seen the movie *War Games*, there was a line spoken in that movie that best exemplifies what this paper is attempting to achieve. When Matthew Broderick was logging into an unknown computer system, he indicated that he would ask the computer for help. He stated that, the more complex a computer system is, the more it has to provide help. SAP provides all types of help for its delivered system. Within SAP you can press F1 on a field to obtain an explanation and purpose of it. Also offered is the CD-ROM documentation which is an extension of their system.

SAP provides a method for documenting company specific information. This paper will focus on three areas that allow for this and are listed as follows:

- Change Requests
- Program Documentation
- Transport Requests

The object will be to describe what can be entered in each of these areas and why you would consider this. In addition, examples of each area will be displayed and explanations will be given on how to create and maintain these documents.

CHANGE REQUESTS

Prior to creating an ABAP program, there is usually a script which contains the requirements of a business function. This script is translated by a technical programmer into a program for the business users. When the programmer creates a new program (or changes an existing one) and attempts to save it for the first time (as other than a local object), the SAP system prompts the programmer for directions before saving the program. The programmer indicates to the SAP system whether to create a new transport/change request or use an existing one and attach this program to it. The transport/change request then captures all changes associated with this program until such time as the programmer decides that the work is completed and then decides to release the program from the development system to the production system for the business users to utilize. Prior to, releasing the program, the programmer should take the opportunity to provide documentation concerning various aspects of what was worked on. The following is a list of what should be provided:

- Script ID and Title
- Script Overview
- Purpose of Request
- Identification of Process Involved
- Issues
- Programming Items to Be Aware Of
- Test Notes
- Modifications Requested
- Modifications Performed

The script ID and Title comes from the request to be processed by the programmer.

The Script Overview should identify the business owner, their priority level, the script type (report, interface, etc.), the date required, who identified the requirement, who created the script along with the date created, the programmer, the date the script was received by the programmer, the date the programmer started work on it, the date the programmer completed the work, who reviewed the work, the date it was reviewed, who approved the work and finally, the date it was approved.

The Purpose of the request should identify the business function to be created.

The Identification of the Process Involved should indicated what components where created or used, the selection criteria, where this information it uses can be found and what you will do with it.

Issues should contain questions about directions to take, unresolved items or future possibilities. This area can be updated as needed with questions for the business owner to answer.

Programming items to be aware off should identify the special steps taken, if any, that are contained in the program. An example could be an interface program where the data format in the sending record for a specified field is packed, and the program had to unpack it prior to processing it.

Test Notes should identify the type of testing that was performed, the components involved, the problems encountered, the resolution of any problems, and the end results.

Modifications Requested and Modifications performed are usually entered after the initial documentation has been completed and transported to production.

In general, change request documentation should describe information from the script. An ID number and business script number along with date information to show various milestones is needed. Also, the nature of the request by the user can be recorded. The programmer can describe what steps were taken to make this occur, and then document any test notes, discussions, decisions, dates, problems, progress, recommendations, etc., that are pertinent to the Change Request. All change requests pertaining to a specific program can be located by looking up the program in the ABAP EDITOR and then selecting VERSION MANAGEMENT. A complete history is readily available of all the change requests regarding that program.

The following is an example of a fully documented change request:.

TA DEVK901018

COACH SCRIPT ID & TITLE

SDIF0200 - SAP Corporate Credit Card to IBM BATCAVE

SCRIPT OVERVIEW

Business Owner:	Bruce Wayne	Priority Level	: Low
Script Type	: Interface (Outbound)	Date Required	: 25 Sep 96
Identified By	: Bruce Wayne	Date Identified	: 18 Sep 96
Created By	: Bruce Wayne	Date Created	: 25 Sep 96
Reviewed By	: Dick Grayson	Date Reviewed	: 6 Nov 96
Approved By	:	Date Approved	:
Programmer	: James Lancello	Date Received	: 23 Oct 96
Date Started	: 28 Oct 96	Date Completed	: 5 Nov 96

PURPOSE OF REQUEST

To collect Corporate Credit Card Information from SAP at the end of each business day and transmit it to the IBM BATCAVE system.

IDENTIFICATION OF PROCESS INVOLVED

- This request identified UPI as the tool to perform the extraction
- An Outbound program was created called ZVUO0001
- Four record structures were created as follows:
 - A. Chrg - Contains the credit card charge information
 - B. Adr1 - Contains customer address information
 - C. Adr2 - Contains additional customer address information
 - D. Btch - Contains the total amount and record counts
- Based the retrieval on the logical database VF for Billing Documents using the Date Created field for the initial extraction
- Additional selection criteria was made using the following order types:

ZCA - Corporate AMEX	ZCV - Corporate VISA
ZCJ - Corporate JCB	ZCR - Corporate Rush Order
ZCRA - Corporate Rush AMEX	ZCRV - Corporate Rush VISA
ZCRJ - Corporate Rush JCB	ZRE - Return Order
ZRPA - Return Order AMEX	ZRPM - Return Order MC/VISA
ZREB - Return Order w/Credit Block	
- Information is then sorted by Order Type and written to the extract file called BILLING.DAT which is currently saved to the workstation of the user running the UPI outbound program
- The file is downloaded to a floppy disk and then uploaded using RUMBA to get onto the mainframe, at this point the programs COE561P, which is added to CLOE160D

ISSUES FOR CLIENT

Should a file be created in UNIX and an FTP performed to transmit the file to the IBM Mainframe

PROGRAMMING ITEMS TO BE AWARE OFF

In order to make this program map correctly to the IBM Mainframe, PACKED DECIMAL FIELDS used the UNPACK command to convert the data to a Character field before being written to the file. Also, in order to get text header information off of the billing record, the function READ_TEXT was used for extraction and mapping as opposed to a direct read of the table STXL field CLUSTD which returned unrecognizable data.

Test notes

With the exception of the charge amount field, the first file created mapped without any problems. The only issue, the amount field was then resolved in future creations by using the UNPACK command to convert the amount field from a PACKED DECIMAL field to a CHARACTER field.

MODIFICATIONS MADE AFTER COMPLETION

Alfred, the loyal butler, added two additional order types for return orders for credit cards. The order types are ZRPA & ZRPM. The UPI program was modified on 11 February 1997.

PROGRAM DOCUMENTATION

When a new program is created for the business the programmer has an opportunity to provide additional information. Properly documented, these programs can provide information for end users, operators, and other programmers. The information entered should contain the following:

- Purpose of the program
- Input required
- Output generated
- How to handle problems
- Operational Issues
- Unix Issues

The purpose of the program should contain a brief description of what this program is intended to do. It should indicate if it is a report or an update program. If it is a report, is it interactive or does it generate a list. The input required refers to the selection criteria. Options should be explained.

The output generated should indicate whether a report will result or a file generated, etc.

How to handle problems should indicate what steps to take in the case of system errors, what to look for if data is incorrect, etc.

Operational issues should guide the data center operators when the program is contained in a batch job and the program cancels as to what steps to take to re-start the program prior to contacting the programmer on call. In some cases the steps described may be enough, if not, that why there is a call list. The main idea behind this is to recognize that there is a problem reduce the response time.

Unix issues should be explained and related to the SAP jobs.

Access for users can be obtained by following the menu path *System -> Services -> Reporting*. From here any program displayed can be selected with a single click, and then select *Go to -> Documentation*.

The following is an example of the program documentation:

RE ZVUO0001

Short text

SAP Corporate Credit Card to IBM BATCAVE

PURPOSE OF THIS PROGRAM

This program was designed to extract Corporate Credit Card Information from SAP using UPI for transmission to the IBM BATCAVE system at the close of each business day.

INPUT REQUIRED

This program requires that the you enter the billing date in the VBRK0001 field in the screen parameter. Usually this will be the current date since this file will be created at the end of each business day. See operational information for running as a batch job.

OUTPUT GENERATED

This program generates a file called ZVUO0001. It has it saved to the Unix directory "/CLIENT/interface/PRD/transfer/OUT" and has a time stamp attached to the file name containing the Century, Year, Month, Day, Hour, Minute and Second (_CCYYMMDDHHMMSS). It can be viewed in SAP by going to the HEALTHBENCH in UPI and using the performing a FILE DISPLAY. You should then enter the Reference Number generated to view the data in the file. The file will be automatically sent by the UNIX daemon to the BATCAVE system on a daily basis except weekends and company holidays.

WHAT TO DO IF A PROBLEM OCCURS

- If the system crashes in the middle of a run, rerun the UPI program with the billing creation date required.
- This program currently uses the following order types to determine which billing records to choose for transmission:

ZCA - Corporate AMEX	ZCV - Corporate VISA
ZCJ - Corporate JCB	ZCR - Corporate Rush Order
ZCRA - Corporate AMEX	ZCRV - Corporate Rush VISA
ZCRJ - Corporate Rush JCB	

If there is a change to order types the program will need to be updated in order for the extract to be correct. In addition, it will not select records for orders less than \$1.

OPERATIONAL INFORMATION

- This program can be located in JOB OVERVIEW by searching for the JOB NAME: SD-U-DLY-FINAL_BILLING.
- In order to execute this program in the JOB it must have the following programs in the sequence identified:
 1. ZZTVARV - with variant ZVUO0001, which identifies the UPI Outbound file.
 2. ZVUO0001 - with variant ZVUO0001, which identifies the path and the file name to create and transmit to the BATCAVE. Also, the variant supplies the variable TODAY which contains the current business day to identify which billing records to select to the parameter VBRK0001.
 3. ZZOUTSEM - with variant ZVUO0001, which creates a semaphore file which tells the UNIX daemon to transmit the file to the BATCAVE and also send a copy to the ARCHIVE_OUT directory

in UNIX.

- If the job fails for any reasons, you can copy the JOB identified above, rename it to CORPORATE_CREDIT_CARDS_TO_IBM_BATCAVE, delete all the programs but the three identified above, then re-run it, or go to JOB DEFINITION and create a JOB with the same name and enter the three programs listed above, save the STEPS, and enter IMMEDIATELY in the START DATE and let it run until the STATUS flag shows FINISHED and the JOB LOG shows that the job finished.

***** I M P O R T A N T N O T E *****

- This program can only be run after the Billing Due List has been run. This program name is RV60SBT1, it uses the variant CLIENT_DEBIT. Please check to see that this step has completed before re-running this program, if it becomes necessary to do so.

UNIX HANDLING OF OUTPUT FILE

- If the program does not complete, then the Unix file will have to be created. Once it is available, the daemon will locate and transmit it to the BATCAVE system and also archive it.
- If the program runs late, inform the Operational staff in Gotham City that the file will be arriving late and to hold up running the BATCAVE update job.
- If the unix file gets lost, locate the file and its corresponding ready file (This will have the same file name with timestamp and will also contain a '.rdy' extension attached to the filename. Copy these two files to the '/CLIENT/interface/PRD/transfer/OUT/' directory so that the daemon will transmit them. This generally takes less than 10 minutes when both files are available. Once the daemon has processed them, inform Gotham City.

TRANSPORT REQUESTS

When the program has been documented and all the programming has been completed, the last remaining step is to complete the transport request. This area should contain information for the basis person who will be making the transport. The programmer should provide the priority of the request, the business area impacted, the specific function, the programmers' name, the business owner, and the individual who approves the transport.

The priority should be: Urgent, ASAP or normal. This allows the approver and the basis person to decide which transports should go first in the case of a heavy load or when under time constraints.

The business area impacted should indicate Sales and Distribution, Finance, Materials Management, etc.

The specific function is the task requested. The programmer, business owner and approver are self-evident.

The following is an example of a transport request.

TA DEVK909031

PRIORITY OF REQUEST

Urgent

BUSINESS AREA IMPACTED

Sales and Distribution

SPECIFIC FUNCTION

Corporate Credit Cards to BATCAVE.

PROGRAMMER

James Lancello

BUSINESS OWNER

Bruce Wayne

APPROVED BY

Dick Grayson

CLOSING REMARKS

In conclusion, let it be said that if all the above information was being provided as the standard, rather than the exception, productivity would be up. Everyone would only have to be trained to use one standardized method for gathering information. With information at our fingertips, we could make faster decisions, respond in less time and raise our productivity and competitiveness and cut costs. Everyone wins, and in the end, that is what we are all after.