SAP Interface Design

Document Prepared By
Timothy Yates

DATAXSTREAM
Systems Integration & Architecture

Revision 2.0

Overview

While there are many different ways to approach SAP system integration design, there is no right or wrong way necessarily, but success usually lies in the attention to details. The following document outlines a possible approach to an SAP integration project. Previous integration knowledge in a particular functional area is beneficial however overall integration experience is more critical. The more hours spent in general on SAP integration the better. Integration technology is continuously changing and adapting therefore continuous learning and training is required to be effective at integration design. This document is broken into 3 main parts:

- Design Considerations – This section discusses critical items required to be successful in an integration project. It is system independent.
- Available Technologies – This section discusses available technology within the SAP system.
- Recommended Design Approach – This section outlines an approach for designing effective interfaces.

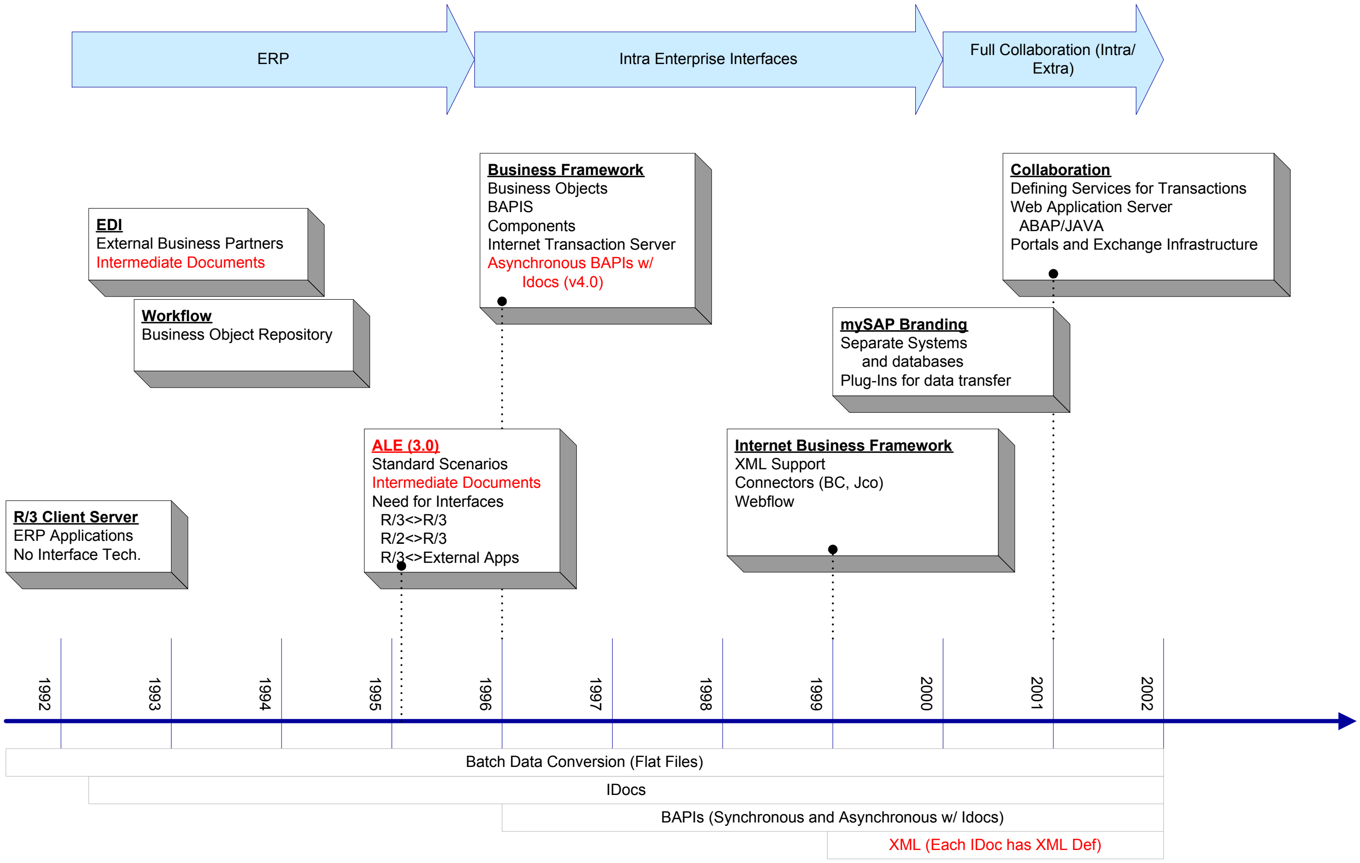
Design Considerations

There are many variables to consider when developing SAP interfaces. It is often impossible to address everything with absolute detail so it is important to understand what things should be considered, identify the important ones, and address them with the most detail. The bigger items are usually easier to see up front. Don't get lost in details that are unimportant because a development process dictates it needs to be address.

- Interface Big Picture
  - Understand Available and Future Architecture and Tools
  - Understand Current Integration Scenarios in Place
  - Assess Resources Available; Skills and Limitations both Functional and Technical
  - Assess Alignment With Long Term Integration Goals and Future Architectural Changes
- Functional Design
  - Sending and Receiving Systems
  - Definition of Data Requirements
  - Definition of Data Dependencies
- Volumetrics
  - Initial Load Volumes
  - Frequency of Records
  - Expected Distribution (Hourly, Daily, Monthly, Seasonally)
  - Expected Long Term Growth
- Hardware \ Disk Sizing
  - Hardware Processing Capacity
  - Hardware Storage Capacity
- Scheduling
  - Understand The Impact of New Design on Existing and Future Integration Requirements
- Test Planning
  - Identification of Critical Tests Required

Available Integration Technologies

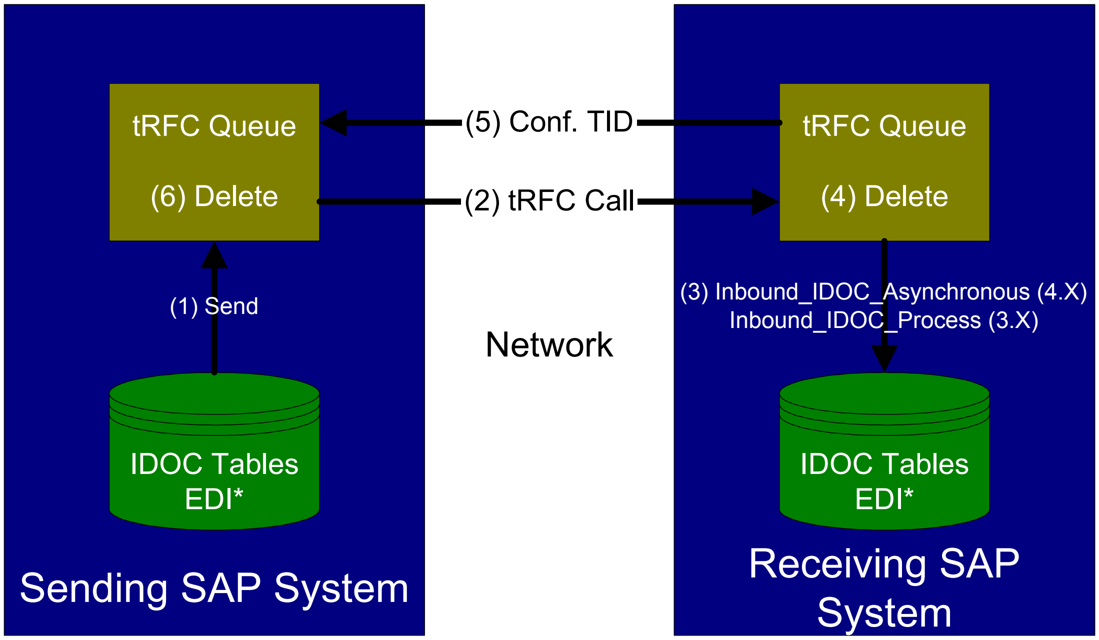
One of the most challenging aspects of interface development is keeping up with changes in technology. SAP's integration technology is continuously changing and adapting, as the technology changes it is complicated further by the increased number of products offered, plugins, and core functionality changes. Below is a chart showing the progression of SAP integration technology of the last 10 years:


ERP
Intra Enterprise Interfaces
Full Collaboration (Intra/ Extra)
EDI
External Business Partners
Intermediate Documents
Workflow
Business Object Repository
Business Framework
Business Objects
BAPIS
Components
Internet Transaction Server
Asynchronous BAPIs w/ Idocs (v4.0)
Collaboration
Defining Services for Transactions
Web Application Server
ABAP/JAVA
Portals and Exchange Infrastructure
mySAP Branding
Separate Systems and databases
Plug-Ins for data transfer
R/3 Client Server
ERP Applications
No Interface Tech.
ALE (3.0)
Standard Scenarios
Intermediate Documents
Need for Interfaces
R/3<>R/3
R/2<>R/3
R/3<>External Apps
Internet Business Framework
XML Support
Connectors (BC, Jco)
Webflow
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
Batch Data Conversion (Flat Files)
IDocs
BAPIs (Synchronous and Asynchronous w/ Idocs)
XML (Each IDoc has XML Def)

The follow table highlights the key features and common issues with various IDOC technologies:

Key Features	Common Issues
EDI / IDOCS	
• EDI technology originally based on file transfer. • IDOCS can be written to files and transferred via standard file transfer protocols (FTP, HTTP, etc) for processing on other systems. • Requires some form of job scheduling to implement. • File interface provides minimal performance overhead when transferring large volumes of data	• Most issues with file interfaces are related to file permissions, file ownership and character conversions moving between platforms. • There are some very old EDI / IDOC interfaces, watch for IDOCS that use workflow for posting they will significantly impact performance negatively.
ALE/IDOCS	
• Asynchronous communication between systems via IDOCS. • Message types (Business Function), IDOCS (Data Container) • Leverages a transactional RFC (tRFC) to guarantee message delivery between systems. • Requires some form of job scheduling: processes controlled via background jobs. • Content based modeling tools, data transformations and change tracking. • Many third party software products and integration products support ALE (tRFC) commumication.	• Added Overhead of tRFC and Data Format (1000 Byte Data Records) • Most issues with ALE interfaces are application\configuration related or with RFC user ID permissions.
ALE/BAPI	
• Leverages BOR (Business Object Repository) – methods. • Can be used for synchronous or asynchronous data transfer. • Most BAPIs are direct input methods (i.e. should be faster). Be sure to test! • Open to non-SAP systems. • Uses content-based data models, transformations and supports a number of change tracking options.	• Similar issues to ALE, field conversion routines (screen fields versus table fields) • Note: In 4.X MOST BAPIS require the commit work to be executed by the calling program.
ALE / Plugins	
• Standard interfaces for tightly coupling SAP R/3 and mySAP products. • Leverages a variety of ALE technologies such as IDOCS and change pointers and tRFC buffers. • Incorporates the use of new BAPIs and BTEs • Supports content-based routing via Variant configuration during model generation (APO CIF Interface). • Uses qRFCs to control dependencies of data object and ensure FIFO processing.	• Performance issues with BTE’s

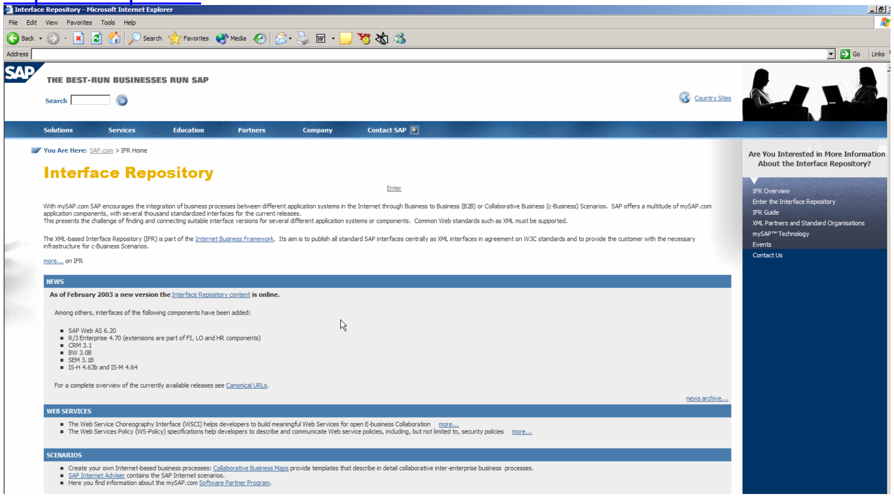
The following diagram shows the steps of ALE communication using the transaction RFC communication layer:



Selecting the correct IDOC interface is important. It is not always clear what interface you should be using and often it is difficult to assess what interfaces are available. SAP offers a number of resources that can help you find what is available standard. Please find below a list of valuable SAP resources to help you pick the correct interface:

SAP Interface Repository

- http://ifr.sap.com


Interface Repository - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links
SAP
THE BEST-RUN BUSINESSES RUN SAP
Search
Country Sites
Solutions Services Education Partners Company Contact SAP

You Are Here: SAP.com > IFR Home

Interface Repository

Enter

With mySAP.com SAP encourages the integration of business processes between different application systems in the Internet through Business to Business (B2B) or Collaborative Business (c-Business) Scenarios. SAP offers a multitude of mySAP.com application components, with several thousand standardized interfaces for the current releases.
This presents the challenge of finding and connecting suitable interface versions for several different application systems or components. Common Web standards such as XML must be supported.

The XML-based Interface Repository (IFR) is part of the Internet Business Framework. Its aim is to publish all standard SAP interfaces centrally as XML interfaces in agreement on W3C standards and to provide the customer with the necessary infrastructure for c-Business Scenarios.

more... on IFR

NEWS

As of February 2003 a new version the Interface Repository content is online.

Among others, interfaces of the following components have been added:

- SAP Web AS 6.20
- R/3 Enterprise 4.70 (extensions are part of FI, LO and HR components)
- CRM 3.1
- BW 3.0B
- SEM 3.1B
- IS-H 4.63b and IS-M 4.64

For a complete overview of the currently available releases see Canonical URLs.

news archive...

WEB SERVICES

- The Web Service Choreography Interface (WSCI) helps developers to build meaningful Web Services for open E-business Collaboration more...
- The Web Services Policy (WS-Policy) specifications help developers to describe and communicate Web service policies, including, but not limited to, security policies more...

SCENARIOS

- Create your own Internet-based business processes: Collaborative Business Maps provide templates that describe in detail collaborative inter-enterprise business processes.
- SAP Internet Adviser contains the SAP Internet scenarios.
- Here you find information about the mySAP.com Software Partner Program.

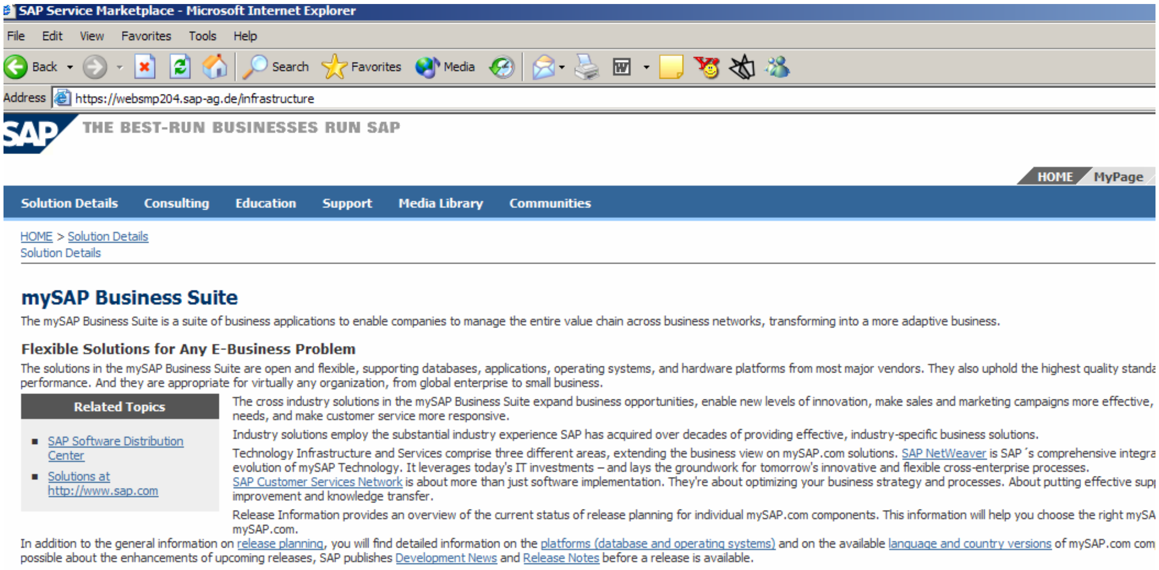
Are You Interested in More Information About the Interface Repository?

- IFR Overview
- Enter the Interface Repository
- IFR Guide
- XML Partners and Standard Organisations
- mySAP™ Technology
- Events
- Contact Us

- There is a lot of useful information related to interface creation on this site.

SAP Services Site

- http://service.sap.com
- This site requires an OSS ID and has bottomless SAP information. Searching this site for specific topics related to IDOCS
- Useful Quick Links
  - Infrastructure
  - Connectors
  - Performance


SAP Service Marketplace - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://websmp204.sap-ag.de/infrastructure
SAP
THE BEST-RUN BUSINESSES RUN SAP
HOME MyPage
Solution Details Consulting Education Support Media Library Communities

HOME > Solution Details
Solution Details

mySAP Business Suite

The mySAP Business Suite is a suite of business applications to enable companies to manage the entire value chain across business networks, transforming into a more adaptive business.

Flexible Solutions for Any E-Business Problem

The solutions in the mySAP Business Suite are open and flexible, supporting databases, applications, operating systems, and hardware platforms from most major vendors. They also uphold the highest quality standa
performance. And they are appropriate for virtually any organization, from global enterprise to small business.

Related Topics

- SAP Software Distribution Center
- Solutions at http://www.sap.com

The cross industry solutions in the mySAP Business Suite expand business opportunities, enable new levels of innovation, make sales and marketing campaigns more effective,
needs, and make customer service more responsive.

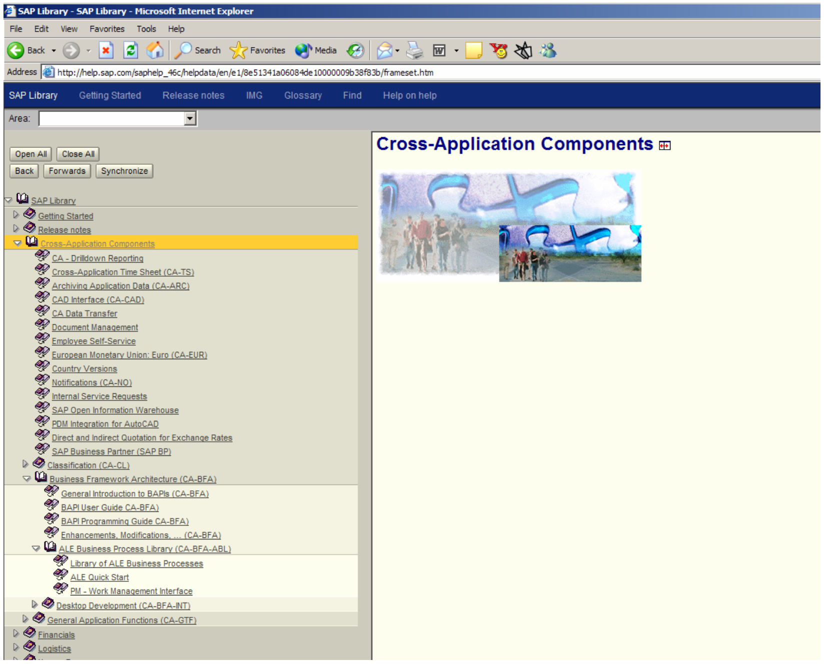
Industry solutions employ the substantial industry experience SAP has acquired over decades of providing effective, industry-specific business solutions.

Technology Infrastructure and Services comprise three different areas, extending the business view on mySAP.com solutions. SAP NetWeaver is SAP ´s comprehensive integra
evolution of mySAP Technology. It leverages today's IT investments – and lays the groundwork for tomorrow's innovative and flexible cross-enterprise processes.
SAP Customer Services Network is about more than just software implementation. They're about optimizing your business strategy and processes. About putting effective sup
improvement and knowledge transfer.

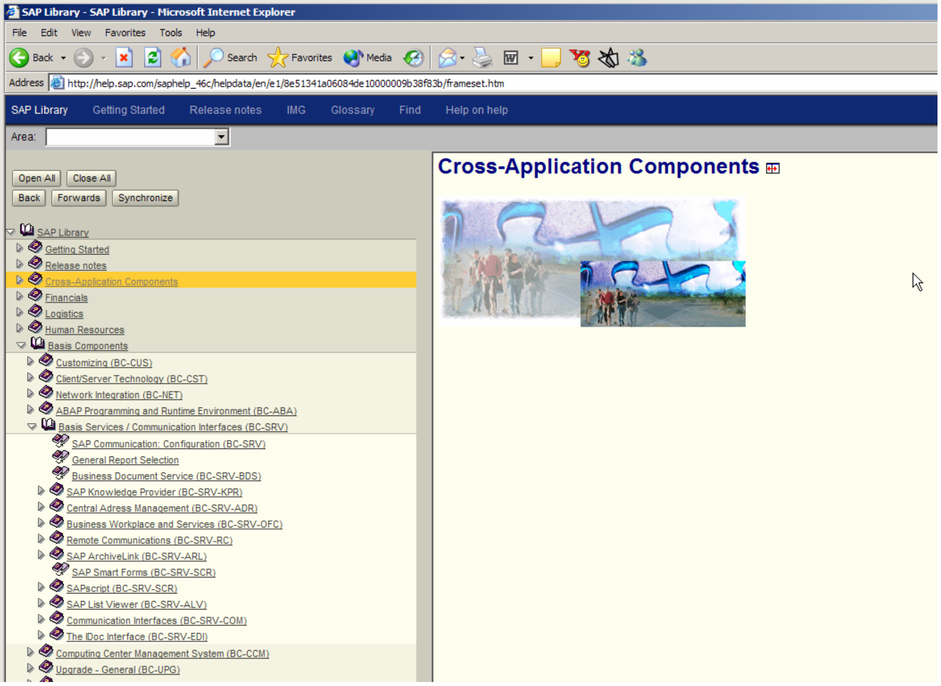
Release Information provides an overview of the current status of release planning for individual mySAP.com components. This information will help you choose the right mySA
mySAP.com.
In addition to the general information on release planning, you will find detailed information on the platforms (database and operating systems) and on the available language and country versions of mySAP.com com
possible about the enhancements of upcoming releases, SAP publishes Development News and Release Notes before a release is available.

SAP Online Help

- http://help.sap.com
- Check: Cross-Application Components >> Business Framework Architecture >> Library of ALE Business Processes

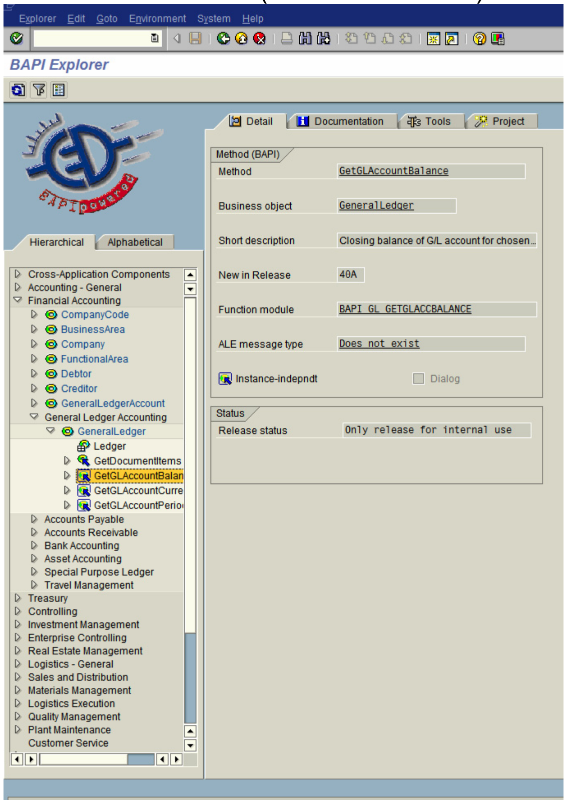

SAP Library - SAP Library - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://help.sap.com/saphelp_46c/helpdata/en/e1/8e51341a06084de10000009b38f83b/frameset.htm
SAP Library Getting Started Release notes IMG Glossary Find Help on help
Area:
Open All Close All
Back Forwards Synchronize
SAP Library
Getting Started
Release notes
Cross-Application Components
CA - Drilldown Reporting
Cross-Application Time Sheet (CA-TS)
Archiving Application Data (CA-ARC)
CAD Interface (CA-CAD)
CA Data Transfer
Document Management
Employee Self-Service
European Monetary Union: Euro (CA-EUR)
Country Versions
Notifications (CA-NO)
Internal Service Requests
SAP Open Information Warehouse
PDM Integration for AutoCAD
Direct and Indirect Quotation for Exchange Rates
SAP Business Partner (SAP BP)
Classification (CA-CL)
Business Framework Architecture (CA-BFA)
General Introduction to BAPIs (CA-BFA)
BAPI User Guide CA-BFA)
BAPI Programming Guide CA-BFA)
Enhancements, Modifications, ... (CA-BFA)
ALE Business Process Library (CA-BFA-ABL)
Library of ALE Business Processes
ALE Quick Start
PM - Work Management Interface
Desktop Development (CA-BFA-INT)
General Application Functions (CA-GTF)
Financials
Logistics
Cross-Application Components

- Check Also: Basis Components >> Basis Services \ Communication Interfaces >> The IDOC Interfaces


SAP Library - SAP Library - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://help.sap.com/saphelp_46c/helpdata/en/e1/8e51341a06084de10000009b38f83b/frameset.htm
SAP Library Getting Started Release notes IMG Glossary Find Help on help
Area:
Open All Close All
Back Forwards Synchronize
SAP Library
Getting Started
Release notes
Cross-Application Components
Financials
Logistics
Human Resources
Basis Components
Customizing (BC-CUS)
Client/Server Technology (BC-CST)
Network Integration (BC-NET)
ABAP Programming and Runtime Environment (BC-ABA)
Basis Services / Communication Interfaces (BC-SRV)
SAP Communication: Configuration (BC-SRV)
General Report Selection
Business Document Service (BC-SRV-BDS)
SAP Knowledge Provider (BC-SRV-KPR)
Central Adress Management (BC-SRV-ADR)
Business Workplace and Services (BC-SRV-OFC)
Remote Communications (BC-SRV-RC)
SAP ArchiveLink (BC-SRV-ARL)
SAP Smart Forms (BC-SRV-SCR)
SAPscript (BC-SRV-SCR)
SAP List Viewer (BC-SRV-ALV)
Communication Interfaces (BC-SRV-COM)
The IDoc Interface (BC-SRV-EDI)
Computing Center Management System (BC-CCM)
Upgrade - General (BC-UPG)
Cross-Application Components

Available R/3 Resources

- BAPI Documentation (Transaction BAPI)



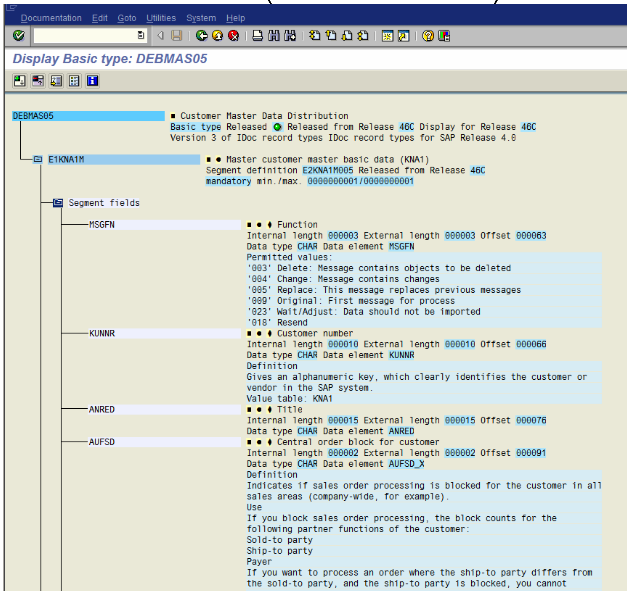
Method (BAPI)

Method	GetGLAccountBalance
Business object	GeneralLedger
Short description	Closing balance of G/L account for chosen..
New in Release	40A
Function module	BAPI_GL_GETGLACCBALANCE
ALE message type	Does not exist
Instance-indepndt	□ Dialog

Status

Release status	Only release for internal use

- IDOC Documentation (Transaction WE60)


Documentation Edit Goto Utilities System Help
Display Basic type: DEBMAS05

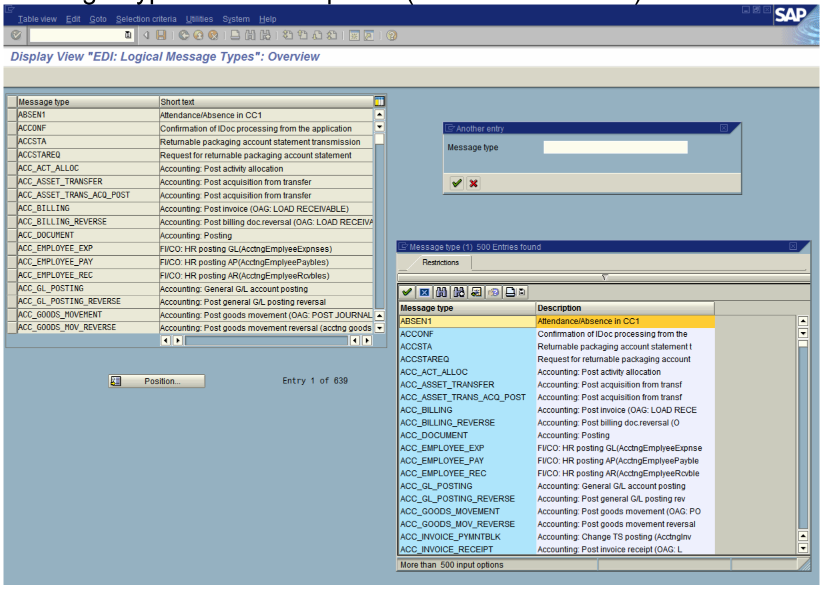
```
DEBMAS05                  ■ Customer Master Data Distribution
                          Basic type Released ● Released from Release 46C Display for Release 46C
                          Version 3 of IDoc record types IDoc record types for SAP Release 4.0

   E1KNA1M                     ■ ● Master customer master basic data (KNA1)
                               Segment definition E2KNA1M005 Released from Release 46C
                               mandatory min./max. 0000000001/0000000001

      Segment fields

         MSGFN                       ■ ● ♦ Function
                                     Internal length 000003 External length 000003 Offset 000063
                                     Data type CHAR Data element MSGFN
                                     Permitted values:
                                     '003' Delete: Message contains objects to be deleted
                                     '004' Change: Message contains changes
                                     '005' Replace: This message replaces previous messages
                                     '009' Original: First message for process
                                     '023' Wait/Adjust: Data should not be imported
                                     '018' Resend
         KUNNR                       ■ ● ♦ Customer number
                                     Internal length 000010 External length 000010 Offset 000066
                                     Data type CHAR Data element KUNNR
                                     Definition
                                     Gives an alphanumeric key, which clearly identifies the customer or
                                     vendor in the SAP system.
                                     Value table: KNA1
         ANRED                       ■ ● ♦ Title
                                     Internal length 000015 External length 000015 Offset 000076
                                     Data type CHAR Data element ANRED
         AUFSD                       ■ ● ♦ Central order block for customer
                                     Internal length 000002 External length 000002 Offset 000091
                                     Data type CHAR Data element AUFSD_X
                                     Definition
                                     Indicates if sales order processing is blocked for the customer in all
                                     sales areas (company-wide, for example).
                                     Use
                                     If you block sales order processing, the block counts for the
                                     following partner functions of the customer:
                                     Sold-to party
                                     Ship-to party
                                     Payer
                                     If you want to process an order where the ship-to party differs from
                                     the sold-to party, and the ship-to party is blocked, you cannot
```

- Message Types and Descriptions (Transaction WE81)



Display View "EDI: Logical Message Types": Overview

Message type	Short text
ABSEN1	Attendance/Absence in CC1
ACCONF	Confirmation of IDoc processing from the application
ACCSTA	Returnable packaging account statement transmission
ACCSTAREQ	Request for returnable packaging account statement
ACC_ACT_ALLOC	Accounting: Post activity allocation
ACC_ASSET_TRANSFER	Accounting: Post acquisition from transfer
ACC_ASSET_TRANS_ACQ_POST	Accounting: Post acquisition from transfer
ACC_BILLING	Accounting: Post invoice (OAG: LOAD RECEIVABLE)
ACC_BILLING_REVERSE	Accounting: Post billing doc.reversal (OAG: LOAD RECEIVA
ACC_DOCUMENT	Accounting: Posting
ACC_EMPLOYEE_EXP	FI/CO: HR posting GL(AcctngEmplyeeExpnses)
ACC_EMPLOYEE_PAY	FI/CO: HR posting AP(AcctngEmplyeePaybles)
ACC_EMPLOYEE_REC	FI/CO: HR posting AR(AcctngEmplyeeRcvbles)
ACC_GL_POSTING	Accounting: General G/L account posting
ACC_GL_POSTING_REVERSE	Accounting: Post general G/L posting reversal
ACC_GOODS_MOVEMENT	Accounting: Post goods movement (OAG: POST JOURNAL
ACC_GOODS_MOV_REVERSE	Accounting: Post goods movement reversal (acctng goods

Position...

Entry 1 of 639

Another entry

Message type

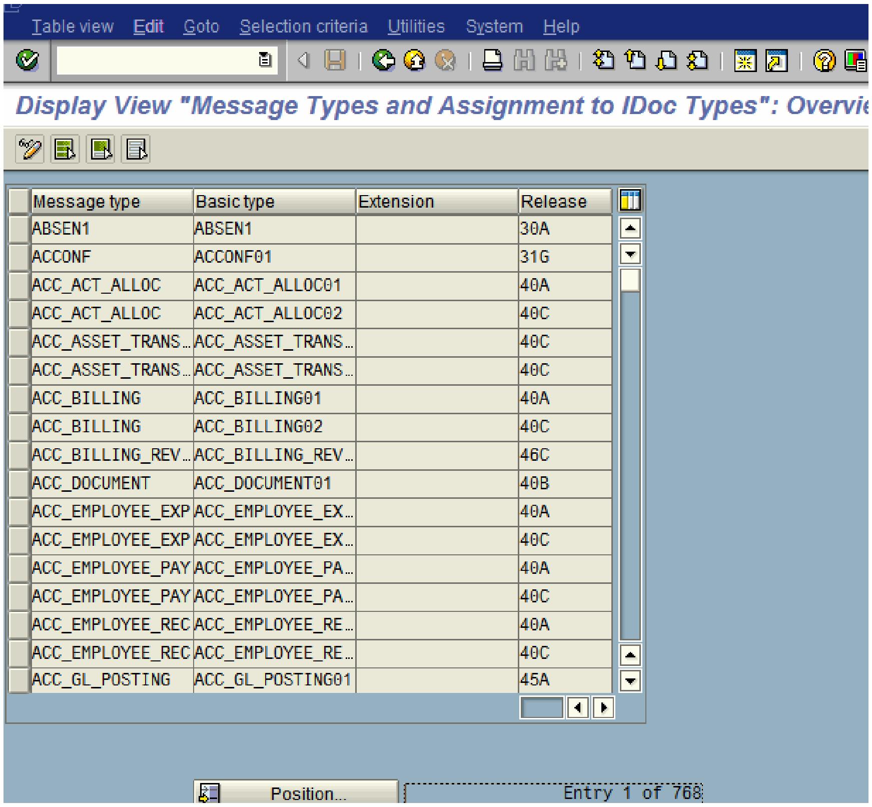
Message type (1) 500 Entries found

Restrictions

Message type	Description
ABSEN1	Attendance/Absence in CC1
ACCONF	Confirmation of IDoc processing from the
ACCSTA	Returnable packaging account statement t
ACCSTAREQ	Request for returnable packaging account
ACC_ACT_ALLOC	Accounting: Post activity allocation
ACC_ASSET_TRANSFER	Accounting: Post acquisition from transf
ACC_ASSET_TRANS_ACQ_POST	Accounting: Post acquisition from transf
ACC_BILLING	Accounting: Post invoice (OAG: LOAD RECE
ACC_BILLING_REVERSE	Accounting: Post billing doc.reversal (O
ACC_DOCUMENT	Accounting: Posting
ACC_EMPLOYEE_EXP	FI/CO: HR posting GL(AcctngEmplyeeExpnse
ACC_EMPLOYEE_PAY	FI/CO: HR posting AP(AcctngEmplyeePayble
ACC_EMPLOYEE_REC	FI/CO: HR posting AR(AcctngEmplyeeRcvble
ACC_GL_POSTING	Accounting: General G/L account posting
ACC_GL_POSTING_REVERSE	Accounting: Post general G/L posting rev
ACC_GOODS_MOVEMENT	Accounting: Post goods movement (OAG: PO
ACC_GOODS_MOV_REVERSE	Accounting: Post goods movement reversal
ACC_INVOICE_PYMNTBLK	Accounting: Change TS posting (AcctngInv
ACC_INVOICE_RECEIPT	Accounting: Post invoice receipt (OAG: L

More than 500 input options

  - Remember to automate the search of this transaction you need to do the follow:
    - Select the Position Button
    - Select <F4> on the Another Entry Pop-Up
    - Use Find Button on Message Type Pop-Up
  - Use the search functionality to find key words, ie (Account, Work Center, Customer)
  -
- Message Type to IDOC Type Mapping (Transaction WE82)



Display View "Message Types and Assignment to IDoc Types": Overvie

Message type	Basic type	Extension	Release
ABSEN1	ABSEN1		30A
ACCONF	ACCONF01		31G
ACC_ACT_ALLOC	ACC_ACT_ALLOC01		40A
ACC_ACT_ALLOC	ACC_ACT_ALLOC02		40C
ACC_ASSET_TRANS…	ACC_ASSET_TRANS…		40C
ACC_ASSET_TRANS…	ACC_ASSET_TRANS…		40C
ACC_BILLING	ACC_BILLING01		40A
ACC_BILLING	ACC_BILLING02		40C
ACC_BILLING_REV…	ACC_BILLING_REV…		46C
ACC_DOCUMENT	ACC_DOCUMENT01		40B
ACC_EMPLOYEE_EXP	ACC_EMPLOYEE_EX…		40A
ACC_EMPLOYEE_EXP	ACC_EMPLOYEE_EX…		40C
ACC_EMPLOYEE_PAY	ACC_EMPLOYEE_PA…		40A
ACC_EMPLOYEE_PAY	ACC_EMPLOYEE_PA…		40C
ACC_EMPLOYEE_REC	ACC_EMPLOYEE_RE…		40A
ACC_EMPLOYEE_REC	ACC_EMPLOYEE_RE…		40C
ACC_GL_POSTING	ACC_GL_POSTING01		45A

Entry 1 of 768

- ○ Use this transaction to compare possible IDOCS candidates, this transaction shows you the number of revisions and the release the revision was available. Use this to eliminate very old interfaces.

Recommended Design Approach
There are many different ways to approach designing interfaces. The following 5 items are critical to a good integration design:

- Requirements
- Assessment and Prototype
- Functional Design
- Technical Design
- Testing

Approaching a design in the following order has some big advantages.

Requirements
Requirements are necessary upfront before any integration project can proceed. The level of detail in the requirements phase can vary drastically: (a) We need to connect SAP to system B. (b) We need to connect SAP to system B, for the following data elements, data dependencies, and business rules. The clearer the requirements are, the better the understanding of the task at hand. A knowledgeable interfaces resource should always review interface requirements and confirm that indeed an interface is required. Often reports are mistaken for interfaces or interface efforts are duplicated in more than one functional area.

Assess and Prototype
Often projects do functional designs next. This is usually a mistake. Getting on the system and identifying possible interfaces available testing initial assumptions and generating initial interface data can significantly improve the functional and technical design. More often than not the functional people writing the functional specification have very little SAP integration experience. Prototyping and developing a basic direction often helps the functional team get the foot hold they need to develop a good functional design. Remember you don’t have to settle with the first thing you try. That is advantage to doing some prototype work up front.

Functional Design
The functional design is the most important part of the process. Starting before you have done any preliminary prototype work almost always means it will fall short of what is required. The functional design needs to address the following items in detail:

- Data Requirements
- Data Dependencies
- Interface Limitations
- Delivery Requirements
- Change Tracking / Triggering Processes
- Volumetrics

Each of these items is important to designing and developing robust interfaces.

Note: Remember there is no perfect design: Get in the Ball Park, No Satellites, No Microscopes. Most designs require some level of changes once the hindsight of production is clear.

- Data Requirements
    - Each system is going to have specific data elements that are required to move from system A to system B. Identify these data elements both at a functional level within the application and at an interface level within the standard interfaces being considered. **Identify the gaps**. This initial review of data elements usually will help in identification of the appropriate interface.
- Data Dependencies
    - Typically the receiving system will drive your data dependencies. Data dependencies require looking at the integration of system A and B at a high level across all required interfaces. While data dependencies exist in individual interfaces they are usually easy to identify if you missed them during the initial unit tests. Data dependency issues across interfaces are rarely found in unit

testing and can also be missed in integration testing, depending on your test plan. There is no cook book approach for finding dependencies, each system being integrated is different. Therefore it is important to understand the systems you are working with. **Look for dependencies when you are developing your initial prototypes.**

- Cross interface dependencies show up as missing data during unit testing. Often developers are working on getting their interface tested so they manually sync the data and lose track of the dependency with another interface. Always try and assess when data is missing during unit testing where you will get that data from in production.

- Interface Limitations
    - Both the sending and receiving systems are going to have certain limitations. It is important to understand these limitations early in the functional design. If specific triggering functionality is required and not part of the standard interfaces being selected than additional functional specification will be required to outline the requirements of a custom based solution. It is also possible that the limitations will create obstacles that can not be resolved. It is better to find them early in the functional design when there is time to adjust overall approach.
- Delivery Requirements
    - How often does an interface need to run? IDOC interfaces are asynchronous, therefore requirements for real-time or synchronous interfaces should not consider IDOCS. The closer to real-time IDOC interfaces get the more performance considerations have to be assessed. Delivery requirements should be driven by real business needs. Developers should challenge function / business process people when a delivery requirement seems unreasonable (to short or to long). **Using job scheduling to control interfaces is always a good idea.**
- Change Tracking / Triggering Processes
    - How is the interface triggered what drives the update? SAP employs many different technologies for triggering IDOCS. Here are some of the more common methods:
        - Change Pointers
        - Message Control
        - Application Specific Send Programs
        - BTE – Business Transaction Events
    - The triggering process is just as important as the data itself. Make sure the IDOC technology selected will support your interface triggering requirements.
- Volumetrics
    - There are many factors that require analysis when reviewing interfaces Volumetrics:
        - Initial Load Volumes
        - Frequency of Records
        - Expected Distribution (Hourly, Daily, Monthly, Seasonally)
        - Expected Growth Over Life of Interface

      Each of these items should help to drive your design. Often, especially with new interfaces these numbers can be difficult to obtain in raw format. Typically some analysis is required to develop projections. The more concrete factors that are identified the more accurate the estimates get. Try to identify particular aspects of the volumetrics that could potentially create issues and design and test for them up front.
    - SAP uses various different technologies in its standard integration interfaces:
        - Call Transactions
        - Direct Input
        - BAPI
        - Workflow

      Make sure that the IDOC you select at a minimum uses a technology that will accommodate your expected loads.

<u>Technical Design</u>

If requirements, prototypes and functional specifications have been done properly the technical design phase should be easy. Implementing what has been specified should be straight forward. The most difficult part of the technical design is knowing what options are available to you when implementing the design. It is always a good idea before you get too far into a technical design to check your approach with other developers. Often having more than one persons input helps improve a design. SAP incorporates a significant number of different interface technologies into its product offering, make sure that you are leveraging the most effective technology and using it correctly to accomplish your design. Also remember that designs that keep as much standard functionality in place typically upgrade with less problems. Don't take a short cut now that will cost you double later.

Testing

Testing is one of the areas that many interface projects fall short. Often test teams develop testing methodologies that are often not applicable to interfaces or do not adequately address the needs of interface testing. There are 4 major testing cycles that should be considered for every interface design:

- Unit Testing
- String Testing
- Integration Testing
- Stress / Performance Testing

Unit Testing

Unit Testing is often performed by the in interface developer. As they are designing a new interface the developer will typically run a component of an interface over and over again testing to see that the particular piece performs to specification. Typically this testing helps to identify that base functionality and the logic of an interface is working as designed. At this stage in the process it is important not to get tunnel vision with your test data. It is usually a lot of work to stage test data, don't fall into the trap of using test data that only exercises 10% of the total interface functionality. Try and cover as many data scenarios early in the unit testing process. Also try to use user ID's with plenty of access there is no value in developers fighting security problems before they have successfully tested their interface components functionality. It is not possible to catch all interface problems in unit testing so expect to make revisions as testing progresses.

String Testing

String testing takes interface testing one step further. Instead of manually staging data and testing individual components of an interface it tests the components functioning together. Often string testing still requires some level of manual intervention between interface components, but is intended to start and tie critical interface components together to see that they function when working together. String testing should occur in 2 steps. The first step should again utilize open user ID's and strictly test interface function. The second step should introduce user ID's that are restricted with production profiles. Often this second step will require several iterations with authorization profiles. This applies both to SAP and other components of the interface design. Here is a list of some of the common issues with security when testing interfaces:

- Wrong User ID or Password. This happens quite often.
- No authorization to specific SAP Function Groups or Transaction.
- No write authorization to file system.
- Write authorization to a file system but no authorization to a specific file on that file system.

Once you have successfully stringed your interface components together and tested your authorizations you are ready for integration testing.

Integration Testing

Integration Testing should replicate production scenarios as close as possible. Do everything possible to setup integration testing like the real thing in production: Use production ID's and profiles, schedule all automated processes, and simulate production loads not only for a single interface but all interfaces. The first two tasks are easy to achieve, getting production ID's and scheduling all automated processes is typically not that difficult, but don't underestimate there importance. Once interfaces are truly running end to end new security issues pop up as well as automation issues. Common security issues in integration testing that arise:

- Wrong User ID or Password. It happened again.
- Automated processes execute typically using different ID's often this can create security issues, both in and outside of SAP.

Common automation issues include:

- Files triggering interfaces before the file is completely written. (Common in Unix)
- Interface job steps running before SAP update processes complete (Common when using the SAP job scheduler)
- Job variants in SAP are incorrectly defined and therefore do not process data correctly.
- Interface jobs running over themselves during loading conditions (Common when using the SAP job scheduler)

Simulating interface production load can be very difficult. While it is not that hard to really load down a particular interface it is not easy to simulate other interfaces running at the same time.

- Interference with other interfaces or other system activities is the number one testing issue missed in integration testing.

Stress Testing and Performance Testing

These two test activities are typically grouped into the same set of test scenarios. The goal of Stress Testing and Performance Testing is often not clear. Often people think of stress testing as loading up a system till it breaks, but what exactly does this prove. It is guarantied that if you want to break an interface by over loading it you can. This as a goal is typically missing the point.

Stress Testing and Performance Testing can also be significantly impacted by other system processes. If you kick off an interface on a system that is also running an Oracle online backup you are pretty much guaranteed to get varying results from test cycle to test cycle.

It is often common for test systems to be configured from a hardware perspective very differently. So what conclusions can you draw from tests performed on different hardware configurations.

So what should stress testing and performance testing accomplish?

Stress / Performance testing should have several goals in mind:

- Establish a baseline for comparison.
  - This includes comparison to projected production interface volumes.
  - A baseline for measuring the impact of parameter changes to different interface components. Prove with numbers that a change had a positive impact.
  - A baseline for measuring the impact of system improvements or upgrades to software components. (Remember upgrading a system does not guarantee better system performance. Often software upgrades required retuning)
  - Provide a point of reference for evaluating the capacity of larger downstream production boxes.
- Identify interface bottlenecks and the need for dampeners within a design.
  - Remember identifying and fixing a bottleneck typically results in a new bottleneck, ie the next slowest process in the integration design. Look for bottlenecks or architecture issues that can negatively impact multiple interfaces.
  - Often high load conditions on certain interface components such as machine to machine communication can cause interface instability. Look for connection bottlenecks and adjust designs to minimize communication load.
  - It is always good to maximize thru put on the SAP tRFC buffer. If queuing is required make sure it occurs in a tool specifically designed for queuing such as IBM Websphere MQ.
- Establish a working expectation for how an interface should perform, both from a performance and stability perspective.

How do you approach it?

- The most important aspect of performance and stress testing is understanding what to measure and how to measure it. What are you going to measure?
  - Thru-put

    - Wait-time
    - End to End Cycle Time
    - System Load
        - Memory
        - Swap
        - CPU
  - Once you have established what you are going to measure how are you going to measure it. It pays to check around, often project have quite a few tools for measuring performance that you might not know about. Here are some possible measurement options:
    - Stopwatch
    - System Monitoring Tools
    - Time Stamps on Files or Jobs
    - System Reports

Performance / Stress Testing is difficult to script and sometimes can be difficult to measure. It is almost always an iterative process, and often does not have a specific goal you can put your finger on. The process usually requires analysis and adjustment to the game plan as testing results are analyzed. In the end it should lead you to understanding your capacity at present and point you at possible areas of improvement.