

The Multiple-Client Concept of the R/3 System

Overview

The term “client” in the R/3 system is a business management organizational term that is defined as “a legally and organizationally independent subscriber to the system”, in which “all business management data is protected against access by other clients” (c.f. client definition in the R/3 Customizing IMG). The client appears on the logon screen of an R/3 system as a qualifier which effects the division of the physical R/3 system into several logical user systems, each of which contains its own data and can be operated fully independently of the other logical systems.

Given this definition one might expect that an arbitrary number of quite independent clients could be operated in parallel in the same physical system (“multiple-client operation”). An initial cost analysis might make this option seem attractive, since it would enable the costs of both hardware and system maintenance to be substantially reduced.

However, every R/3 application requires certain central resources which are used for implementing the basic functional elements of the software concept. These include the complete ABAP/4 Repository and also a number of Customizing settings which for good reason have to be client-independent. These will be described in more detail later in this report.

In view of these facts, the multiple-client concept of the R/3 system appears more questionable, particularly to the extent that the existence of each client could cause (uncoordinated) changes to be made to the central client-independent resources. It is clear that there must be at least some rules and boundaries governing multiple-client operation. Those rules will impose certain restrictions on the administration and development of individual clients. But they will be needed for the proper, orderly operation of a multiple-client system.

This report considers all factors affecting potentially critical data and metadata in the client-independent Customizing and the ABAP/4 Repository. It shows how this data can be handled in multiple-client operation given conflicting requirements, and suggests how rules and recommendations for system organization may be derived.

The aim is to equip the system planner with the knowledge he needs to be able to weigh the cost advantages of multiple-client operation against the more flexible administration afforded by single-client operation and to make an competent decision.

Terminology

Multiple-client capability

This refers to the property of a system which makes it possible for it to contain several clients working in parallel and isolated from one another in the following respects:

- ❑ the data of each client is protected so that other clients can neither read nor change the data (data isolation).
- ❑ two clients are able work in parallel without disrupting each other or affecting each other in any way (functional isolation).
- ❑ the client configuration of a system may be changed at any time, for example by adding a new client, or by removing or changing the Customizing settings of an existing client (independence).

Data isolation

The SAP system provides full multiple-client capability at the level of the application data. Any data belonging to a given client is visible only within that client. It cannot be changed or displayed from another client using application functions. There are, however, specially designated utility functions for the system administrator which allow read-only access, in order to copy a client or to reorganize the database tables, for example.

Functional isolation

The functional aspects are rather more critical. Every operational system contains two kinds of functional element. On the one hand there are the general (and therefore client-independent) standard R/3 functions, but on the other hand there are also the client-specific functional elements which are introduced by Customizing for modeling specific business processes.

- ❑ **Standard functions in the R/3 system**
These are the functions of the R/3 system that are represented by the program objects of the ABAP/4 Repository. These constitute the basic functions involved in any usage of the R/3 system and thus also of each client. The client-independence of the complete ABAP/4 Repository together with its program and dictionary objects is a fundamental characteristic of the R/3 system.
- ❑ **Functional enhancements resulting from Customizing**
Before a client can become operational, its organizational structures and business processes have to be modeled in Customizing. The majority of the control data that have to be maintained are client-specific and are therefore not of relevance to this report. Some of the control features to be modeled have Repository character and may lead to programs and Dictionary elements being generated. In this case they supplement the standard set of SAP functions delivered and are also client-independent ABAP/4 Repository objects. The relevant generation information is also stored as client-independent data in Customizing tables.

Independence and system operation

Clients for testing and training purposes do not usually present a problem, since they are a copy of a client already existing in the system. They are thus based on the same Customizing and the same Repository (program objects). These clients are therefore not independent in the sense described above but may, at least from a functional point of view, be seen as a derivative of the client from which they were created.

However, where there are several independent companies which are to be modeled in separate clients on a single physical system, the above is no longer the case because of the client-independent objects which have to be introduced.

The fact that client-independent functions may be created as a result of Customizing has certain consequences for a multiple-client system. Given the necessary authorizations, client-independent Repository objects are accessible from every client. This means that there is a certain danger that a client's settings and enhanced Repository objects could be accidentally changed or overwritten by another client. An added problem is that this often cannot be detected by testing because of the data isolation of the client involved.

There is therefore a basic conflict between the need for each individual client to be independent from each of the others and the cross-client nature of the ABAP/4 Repository. In order to guarantee the integrity of each client of a multiple-client system in this environment, the SAP system administrator must himself take on the full responsibility for the maintenance of the client-independent objects. The best approach to the maintenance of these objects is to organize it centrally in a specially designated client, with carefully matched authorizations for the ABAP/4 Workbench and an agreed naming convention for the additional objects created in each client.

The question of multiple-client capability is therefore closely connected with the question of the practicality of organizing this kind of central maintenance. This kind of maintenance is obviously straightforward if most of the problems can be dealt with in the Customizing and maintenance phases. It will, however, be hard to organize in the case that requirements constantly arise out of the normal operation of each client and needed to be answered at a central point.

In order to be able to answer this critical question we will first outline the set of Customizing objects. There follows an analysis of the effect of client-independence on the operation of clients. The last part of this report, entitled "Description of the client-independent Customizing objects", explains for each of the Customizing areas of the groups B and C how diverse requirements of clients can be addressed in multiple-client operation.

Why is client-independent Customizing needed?

When the number of applications with highly customer-specific characteristics grows then it becomes difficult to design and implement all system functions as standard software. Examples of this are to be found in areas such as pricing in SD, profitability analysis in CO-PA and the executive information system EIS. These applications typically consist only of framework software together with a 'toolkit' for tailoring and completing the application according to the customer's needs. The latter uses generation techniques from Customizing, i.e. the "Customizer" actually performs the work of a software developer, although with computer-support and at a high, near-application level. This approach makes it possible to cope with individual, non-standardizable application profiles flexibly without compromising on standardized, easy-to-use user interfaces.

The set of client-independent Customizing settings may be divided into three categories:

- A. Technical settings, relating to the customer's technical environment and system profiling.
- B. Repository-relevant system adaptation: This mainly includes automatic generation of parts of the application, so these settings have program character.
- C. Globally valid settings that can be equally relevant for any of the clients.

All client-independent objects contained in the Customizing (-IMG) will now be considered, divided into these three categories.

A. Technical Settings

This category includes the settings used for tuning the system to the available resources, for technical configuration and the settings of the system profile parameters. These settings have no effect at all on the business management functions of the applications.

- The following are representative of this category:
 - (a1) RFC destinations
 - (a2) Technical network installation
 - (a3) Communication control
 - (a4) Configuration of the archive management and Archive Link
 - (a5) Communication control for EDI and ALE
 - (a6) Platform-independent filenames for applications
 - (a7) Client definition
 - (a8) Definition of customer text types
 - (a9) Parameters and settings for the ABAP/4 Workbench

B. Repository-Relevant System Settings

This category includes settings that are introduced into the system early on in the Customizing, usually immediately after the installation or immediately before new applications go operational. They are used for modifying and enhancing the Repository meta-definitions. They form an important part of the foundation on which subsequent Repository objects are generated (programs, interfaces, Dictionary definitions), which are themselves used for tailoring complete applications.

- The following are representative of this category:
 - (b1) Definition of conditions
 - (b2) Definitions from LIS (Logistics Information System)
 - (b3) Enhancements to the account assignment block
 - (b4) Setting up FI-SL table groups (“Special Purpose Ledger”)
 - (b5) Validation, substitution and rules
 - (b6) Report Writer and Report Painter
 - (b7) Assignment groups in the profitability analysis (CO-PA)
 - (b8) Global definitions in the EIS and in the profitability analysis (CO-PA)
 - (b9) Definition of fast entry in FI
 - (b10) FI rules for allocations and payment groups
 - (b11) Definition of classifiable objects for the classification system

C. Globally Valid Settings

These settings are generally independent of any particular company and therefore not restricted to any particular client either. Examples include currency definitions, local data exchange regulations for government agencies (such as the German *Datenübermittlungsverordnung*) and tax office codes. This approach helps reduce the maintenance overhead in a multiple-client system, since these settings only have to be maintained once in each physical system and not repeatedly for each client.

- The following are representative of this category:
 - (c1) Data exchange regulations for government agencies and tax offices
 - (c2) Factory calendars and holiday calendars
 - (c3) FI correspondence types
 - (c4) Decimal places for currencies
 - (c5) Central address management
 - (c6) IMG Implementation Guide

Analysis

The difference between categories A, B and C.

The objects listed in category A above (technical system settings) do not have a critical effect on multiple-client operation and will not be further discussed in this report.

The object types listed under B and C have considerably more relevance for the application and therefore their influence on multiple-client operation needs to be considered carefully. The following discussion is concerned mainly with these objects.

A solution involving the definition of several versions of objects

In each of the Customizing areas under consideration it is still possible to model the differing requirements of competing clients despite the need for client-independent definition. The answer is to define several versions of a Customizing object type. These are then addressed via different key terms and allow the individual, client-specific functionality of competing clients to be modeled.

For example, in the case of the conditions tables (b1) a special customer naming space is reserved. This can be used to allow the creation of almost any number of individual conditions tables which may be offered for use to the various clients. Another example is the case of the holiday calendars and factory calendars (c2): It is quite likely that on the one hand several clients will each require their own specific calendar, but also on the other hand that there will be instances where several clients all need to use the same calendar. In order to reduce and simplify administration it would make sense to allocate a calendar to a cross-client geographical or industry-oriented environment, in order to avoid the redundancy of maintaining the calendar in every client.

The central maintenance of client-independent Customizing data

The maintenance of client-independent Customizing objects in a multiple-client system is only possible given certain rules of system organization. On no account may the maintenance be undertaken by the clients themselves, even though this is the normal way of working with client-specific objects in Customizing. It is important that the system administrator can manage the competing requirements on the cross-clients objects centrally and can arbitrate between the various clients where necessary. As explained above, the best way of achieving this is usually to have several versions available which can be allocated to clients as and when necessary.

SAP recommends that you do not generally make significant changes to the Customizing settings directly in the productive client. Changes to the productive system should usually be restricted to exceptional cases. Customizing changes generally have an effect on application behavior similar to that of program changes and should therefore be made first of all in a test or development system, subjected to quality control and then released before they take effect in a productive environment.

It is therefore to be recommended that a test or development system is maintained parallel to the productive system, and that the systems have an identical client structure, i.e. that the client NNN in the productive system is also present in the test system with the same client number NNN. If you also have an additional production preparation system, then the same applies for this system too.

Repository maintenance (modifications and customer developments)

SAP also recommends that any changes to be made to Repository objects - not just those for multiple-client operation - should not be made directly in the productive system, but that a separate test/development system be maintained specially for this purpose. The changes should be copied to the productive system only after quality control.

If the above suggestion is followed and the maintenance of cross-client control data inside a client is avoided, then the system administrator will have to organize some kind of central maintenance. The same applies to the Repository changes. One obvious alternative is to designate a particular client for central maintenance. Using the client control provided in 3.0 (extended table T000), it is quite easy to ensure that all other clients are excluded from Repository changes via special protective measures (see the “protective measures” section). It is of course also possible to implement the restrictions in the form of authorizations. Only very approximate tests can be made in the central client, and testing should therefore be done in parallel in the specific utility clients using the test data available there.

As this short description should have made clear, a multiple-client system is not really suitable if the participating clients make a large number changes to Repository objects or if they are responsible for much new development activity. We therefore recommend a single-client system for “Repository-intensive” clients. It is not possible to make a clear division between “Repository-intensive” clients and others; the deciding factor should be the centralization overhead of doing program development in a multiple-client system, as described above.

System administration

In the area of system administration it is important to remember that the following functionality is not supported at the client level but only at the level of the physical system:

- Reorganization programs
- The backup of data
- Release upgrades.

It is in fact an important aim of data administration to ensure that all such functions are realized using the higher performance utility functions offered by the database and hardware providers.

Summary of the analysis

- The competing requirements of the various clients in a multiple-client system can be modeled in parallel successfully given adherence to certain rules.
- The cross-client Customizing objects cannot be maintained from within any of the individual clients involved. Instead the system administrator must organize central maintenance, which is best done in a specially designated client.
- This also applies when the ABAP/4 Workbench is being used, i.e. when the customer is developing his own applications or making modifications to software.

- ❑ Clients that need to make frequent or many changes to Repository objects are not suitable for multiple-client operation. A single-client system is preferable.
- ❑ Technical maintenance (reorganization, data backup, upgrades) cannot be performed on a client-specific basis.
- ❑ Normal business management operations and processes do not lead to cross-client settings or Repository objects being changed.

Protective measures for multiple-client operation

It will be clear that in all of the multiple-client scenarios under discussion the cross-client settings are susceptible to being accidentally overwritten by one of the participating clients. Therefore a special protection concept is needed to guarantee the smooth coexistence of several clients in a single physical system. Release 3.0 offers three different protective measures, all integrated into the maintenance process of the objects.

- ❑ Indication of cross-client effect.
When any of these objects is maintained a pop-up window appears at the start of the maintenance and warns the user that the changes made will affect all clients in the system.
- ❑ Special authorizations for cross-client maintenance
There are special authorizations for protecting the maintenance of these objects.
For example, in the case of SM30/SM31 maintenance there is a special authorization S_TABU_CLI. A user wishing to perform maintenance must possess this authorization in addition to the normal table authorization.
- ❑ Cross-client maintenance can be limited to certain clients.
Using the central client maintenance (table T000), it is possible to specify for each client whether the client is permitted to make cross-client settings or not (3.0C onwards). Therefore for each client it can be specified individually whether, in addition to the client-specific data, client-independent Customizing data, Repository objects or both can be changed.

These measures fulfill the requirement mentioned above by allowing the definition and administration of cross-client Customizing to be organized centrally and to be protected against unwanted interference and accidental overwriting of central data.

Description of the client-independent Customizing objects

The following section is a discussion of the object types in the client-independent Customizing listed under B and C, and of their influence on the multiple-client concept. Where appropriate, organizational recommendations for multiple-client operation are also made.

(b1) Definition of price setting, conditions

Function:

- ❑ The condition technique is a flexible tool that allows you to define data records with key fields that may be freely selected. The condition technique is used for defining:
 - prices, surcharges and discounts, taxes
 - rebate agreements
 - message defaults
 - account determination
 - material determination
 - material listing and material exclusion
 - batch determination
- ❑ Two of the elements of the condition technique, namely the condition tables and the access sequences, have client-independent definitions and can therefore be used all across the system. One result of defining condition tables and access sequences is that Repository objects are generated (tables, screens and selection reports).
- ❑ Condition tables: The combinations of the various criteria for which prices (for example) can be agreed are specified in the condition tables.
- ❑ Access sequence: The access sequence is a search strategy used by the system for searching for valid data for a particular condition table. It consists of one or more accesses. The sequence of the accesses determines the priorities of the various condition records...
- ❑ You can find more information about conditions in the on-line help for sales and distribution and in the IMG under Sales and Distribution -> Basic functions...

Competing requirements of various clients:

- ❑ Restrictions on the use of condition tables and access sequences in certain clients can be imposed via Customizing by allocating them to a scheme (e.g. pricing schema, message schema,...). This technique allows differing strategies for the above areas of use (e.g. price setting, message default,...) to be specified for each client.

(b2) Definitions from LIS (Logistics Information System),

Function:

- ❑ The Logistics Information System
The logistics, inventory controlling, purchasing, shop floor, maintenance, quality and sales information systems together constitute the Logistics Information System (LIS). Statistical data is automatically aggregated from the individual applications into statistics tables, known as information structures. These information structures form the basis for analyses in the LIS.
In order to supplement the use of the information structures supplied by SAP, one can generate one's own information structures in Customizing, in order to satisfy special requirements. There is one of these self-generated information structures available system-wide, i.e. in all clients.

Competing requirements of various clients:

- ❑ How the statistical data is recorded in a self-generated information structure is configured individually in each client. If an additional information structure is required within a client, then this can be created by the customer - as described above - and made available in the client. The concept allows the definition of up to 500 additional structures. You can find documentation about generating information structures and about the client-specific configuration of the data recording in the Implementation Guide under Logistics general -> Logistics Information System -> Logistics Data Warehouse...

(b3) Enhancements to the account assignment block

Function and operation:

- ❑ The account assignment block contains the collection of all SAP account assignment elements and also of customer-defined additional account assignments. The functionality allows the customer to:
 - include customer-specific fields as 'Customer Includes' in SAP tables
 - define his own sub-screens (naming convention: 9000-9998)
- ❑ Each new account assignment field and each new customer sub-screen is entered in an R/3 system as one of the Repository objects with a system-wide unique identification. Therefore in multiple-client operation the name has to be assigned centrally. The new customer fields will be visible in the database for each client. The data recording is performed via client-specific tables in each client.

(b4) Setting up FI-SL table groups

Function:

- ❑ The SAP customer defines one or more “Special Purpose Ledgers” in the FI-SL application. A “Special Purpose Ledger” (previously called “extended general ledger”) is a customer-defined ledger that can be kept in the application with any account assignment elements as a general ledger or a subsidiary ledger. Accounting values for customer-defined account assignment elements can be managed in FI-SL as summary tables and line item tables for reporting purposes. In addition to the activity-oriented validation, accumulation and collection of data via integration interfaces, periodic functions such as planning, allocation, rollup, currency translation, balance carry forward and data transfer are also available.
- ❑ In order to save Special Purpose Ledger data, a part of the Customizing work the customer carries out will usually be to add new table structures (table groups) according to the customer naming convention to the SAP Standard Repository. For a multiple-client scenario table groups can be allocated to specific individual clients by means of the naming convention. Control table entries have to be present before it is possible to work with a new table group in FI-SL.

Since the SAP Repository is client-independent, some of the FI-SL control tables are also client-independent. The contents of the table that is most important in this context (T800A) describe the dependencies between the individual tables within a table group (FI-SL table directory). Fixed field movements (T800M) contain control information for SAP standard fields (for example, the company code), without which no kind of meaningful data update can take place. User Exits (the programs are defined via T80U) are program components and thus client-independent. The T800A and T800M tables are filled from an FI-SL installation program. In this context it is important to bear the cross-client problems in mind.

- ❑ Transactions are business-oriented outlines of all accounting transactions modeled in the SAP system (T022). Customer-specific FI-SL transactions only need to be created if customer-specific interface programs also will be transferring data to FI-SL, in other words, in the case that the customer has extended the SAP standard with further applications of his own.
- ❑ Beginning with release 3.0 there are also the so-called “standard ledgers” (general ledger, profit center, consolidation, reconciliation ledger) in addition to the Special Purpose Ledgers. These SAP standard ledgers are stored in table groups that customers are not able to change. Updating of these standard ledgers is client-specific.

Competing requirements of various clients:

- ❑ Multiple-client operation in which the various clients have differing requirements means that it may be necessary to define several different FI-SL table groups. These would then be potentially available to all clients. On the other hand the actual run-time update of the data can be controlled on an individual per-client basis, because it is a part of the client-specific Customizing.

(b5) Validation, substitution and rules

Function and operation:

- ❑ Validation is used for defining customer-specific checks for SAP fields, customer fields or combinations of these fields. If any of these conditions is not fulfilled the system issues a warning or an error message; this can be tailored to the customer’s wishes. Thus the user can be informed that his input was incorrect or even prompted to modify his input.
- ❑ Substitution can be used to replace or to add to the input made in specific fields. This technique means that the work of data entry can be substantially reduced.
- ❑ It is also possible to use rules and User Exits when defining validations and substitutions.
- ❑ SAP presets all possible fields in the structures of each application for validation and substitution. The number of fields selected can be restricted for each individual client in the client-specific Customizing.
- ❑ The customer-specific validations and substitutions are stored client by client and are processed in the respective clients.

(b6) Report Writer and Report Painter

Function:

- The Report Writer can be used to report on data from various different applications. The Report Writer selects the data requested and formats it as required. The Report Painter can be used for designing the reports. The use of groups (sets), variables, formulas, cells and key figures allows reports to be created for very specific requirements. The Report Writer is used in several different areas, for example in the areas of indirect costs controlling, LIS and Special Purpose Ledger.

Competing requirements of various clients:

- The definition of a report is client-specific and each client can therefore have a different version of it. The connection between database tables and the Report Writer is, however, client-independent. If a table or information structure (flexible analyses) is connected to the Report Writer, then it is available in all clients with the same semantic properties. The connections between tables and the Report Writer are controlled by the client-independent T804 tables. These tables contain attributes for applications structures that assign further semantic properties to these structures. They refer only to the structure in question, and are therefore quite independent of the current client. For example, if you install a table in the area of the Special Purpose Ledger and connect it to the Report Writer, then this table can be analyzed in every client. If in the Report Writer you add further virtual characteristics to the characteristics of the table (tables T804B, T804C), then these virtual characteristics will also be available in every client.

(b7) and (b8) Global definitions in EIS and profitability analysis

Function:

- In CO-PA the operating concern and assignment group objects are defined cross-client and can be addressed via a 4 character key code. In both areas different objects can be defined for each client. This means it is possible to support competing requirements within a multi-client system.
- In EIS the generated objects of the data basis are defined cross-client. These are the DDIC objects (aspect tables, data elements and domains) with database tables and field catalog entries. Since aspect numbers, field names and all other identifiers are explicitly defined, a well-organized naming convention should be able to satisfy competing requirements.

Competing requirements of various clients:

- The views onto the defined objects can be restricted through the use of client-specific authorizations. Analyses can also be controlled by client. During maintenance a pop-up window provides a warning about the cross-client effects of the change. You can find further documentation in the IMG.

(b9) Definition of fast entry in FI

Function:

- In financial accounting there are special entry screens for the fast list-oriented entry of account line items. Because these entry screens contain only a selection of the more commonly used entry fields, the customer has the option in the settings menu of generating further entry masks containing other fields corresponding to his requirements.

Competing requirements of various clients:

- Entry screens are client-independent Repository objects. They are therefore centrally created and administered.

(b10) FI rules for allocation and payment groups

Function:

- Allocation for item sorting:
Using the rules a report for doing the allocation is generated that ensures that the 'allocation' field in the line item automatically gets filled with the correct contents. Since the report is client-independent, all rules apply equally to all clients....
- Payment groups, payment texts:
In this case too a report is generated using the rules for automatically generated payments. This report ensures that the required grouping characteristic in the processing is taken into account by the payment program. For example, a grouping of the line items according to contract number (loans area) can be achieved. For each contract number a separate debit memo would then be created for the customer...

Competing requirements of various clients:

- Rules have Repository character. They are client-independent and therefore have to be entered centrally in the multiple-client system and must be administered centrally from the point of view of the engineering change management. If an addition is to be made to the standard, then customer-specific allocation rules should always be defined with a rule number above 100.

(b11) Definition of classifiable objects for the classification system

Function:

- This Customizing function allows the user to define objects for the classification. The name of the table in which the objects are stored has to be specified, (e.g. MARA for the material masters). In this Customizing table SAP delivers all object types that according the standard are planned for a classification within the classification system. Customers can add further object types of their own to these standard entries. Any changes made here will take immediate effect in every client in the system. However, the object types are not incorporated directly into the classification; this occurs via the class type, which always refers to the object types.



Further information:

R/3 IMG - Cross-application functions - Classification system - Classes - Object types

Guide to classification system, Chapter 10, New Objects.

Competing requirements of various clients:

- ❑ Object types have Repository character and therefore have to be administered centrally. However, the definition of the class type that refers to an object type is client-specific. Therefore it is possible in Customizing to control the usage of one of these objects in the classification in a way that permits competition between different clients, because the object gets included in the classification only via the class type.

Recommendations for the organization:

In the case that different clients have differing requirements in respect of new objects, then this should be taken into consideration when maintaining the objects. Thanks to the client-specific definition of the class types that refer to the objects, the use of object types that are valid in the whole system can be restricted to specific clients in Customizing.

(c1) Data exchange regulations for government agencies and tax offices

Function:

- ❑ These tables contain key codes that are needed for dealing with tax offices and other government bodies. These codes are of course independent of any particular client.
An example based on the case of Germany: There is a table known as the DÜVO table, used to regulate data exchange in government agencies (DÜVO = *Datenübermittlungsverordnung*), that contains details of the permissible nationalities for DÜVO reports. These nationalities are published by the Federal Statistics Office and apply across the whole of Germany. Changes and additions are either supplied by SAP or the customer can do this maintenance himself for all clients centrally. This ensures standardization across all clients. The German tax office table lists all tax offices that have jurisdiction across the whole country. This information is published as part of the Federal Tax Bulletin.

(c2) Factory calendars and holiday calendars

Function:

- ❑ The “factory and holiday calendar” allows calendars to be defined that are specific to a given company or location. It is based on the public holiday, holiday calendar and factory calendar components. A rule-based definition is used for public holidays. A holiday calendar defines a set of public holidays that apply in a particular region and thus defines the calendar for a specific location (e.g. for the German federal state of Baden-Württemberg). A factory calendar is based on the applicable holiday calendar and extends this by including the specific requirements of an individual company, for example company holiday shutdowns, special shifts, etc.

In order to simplify the customer's settings as much as possible, SAP supplies details of most commonly applicable public holidays and also a collection of holiday and factory calendars. The customer may add further public holidays, holiday calendars and factory calendars to these standard settings. Any changes made here will take immediate effect in every client in the system.

- ❑ Further information: R/3 IMG - Global settings - Maintain calendars

Competing requirements of various clients:

- ❑ Any change to the holiday and factory calendar takes effects across the whole system and in every client in the system. If a number of clients have conflicting requirements, this can be taken care of only by means of the explicit maintenance of the calendar components.
The various calendars available in the holiday and factory calendar system are identified by a holiday or factory calendar ID that is unique across the whole system. Therefore it is possible for a uniquely identified holiday or factory calendar to be defined for each separate client-specific requirement. A client-related assignment of calendar IDs to the various applications (e.g. HR, SD etc.) makes it possible for a client-based structure of holiday and factory calendars to be drawn up..

(c3) FI correspondence types

Function:

- ❑ A correspondence type in FI identifies the contents, the structure and all other Customizing settings for a letter to a business partner that is to be automatically created. Basically this identification can be freely assigned via the settings menu. SAP has defined certain correspondence types as fixed and these apply to all clients.
If further correspondence types are required in addition to those supplied in the standard, these should not begin with the letters SAP. An existing correspondence type should only be deleted or changed centrally because of the cross-client effect of the action.
- ❑ The definition of a correspondence type contains both cross-client and client-specific components. The main cross-client components are the basic properties and relationships to structures in the application (e.g. references to documents). This part therefore has Repository character. Information about usage is client-specific, and this information usually can be specified up to the level of the company code.

(c4) Decimal places for currencies

Function:

- ❑ The decimal place parameters for a currency are a property of the currency itself and are therefore not client-specific. For example, the USD, DEM and GBP currencies have two decimal places, but the ITL, ESP and some others have none. If the recommendation of keeping to the generally accepted ISO norms for the currency keys is followed, then no changes to the supplied table entries will be necessary.



(c5) Central address management

Function:

- 1. Customizing activity: "Maintaining display formats for address versions":
It is possible to maintain several character set versions of an address in the master data of customers, vendors and banks. This requirement exists only in countries that use an extended code page, including Japan (double byte) and East Europe, and it is linked with the code page and is therefore dependent on the installation. The Customizing setting controls which additional character set versions can be maintained in the system parallel to the original version of the address.
- 2. Customizing activity: "Maintaining communication types"
This is a Customizing setting that determines which communication types (e.g. fax, e-mail, X.400, etc.) can be maintained in the central address management and which communication types can be directly selected from the send screen in SAPoffice. The usability of some communication types depends on a special installation. This is not client-specific.

Competing requirements of various clients:

- Typically the maintenance of an international character set is possible in addition to the national version. It is not a problem if this is not actually required in all clients, since a function which is active but not used in a client does not cause a conflict. It is not possible to deactivate the function in individual clients as long as at least one other client requires it.
- For both the address maintenance and sending all communications types that are needed anywhere in one of the clients have to be activated in Customizing. Address maintenance for these communications types can then always be performed in any client. This is redundant, but does not cause a problem. When sending from SAPoffice, however, it is also possible to deactivate individual communications types on a client-by-client basis in the SAPoffice system profile (transaction SO16).

(c6) IMG Implementation Guide

Function:

- The following elements used in connection with the implementation of R/3 are client-independent:
 - Enterprise IMG
 - Project IMG
 - Views on the Project IMGThese elements are used to guide the customer through the steps of system implementation (Customizing).

- ❑ The aim here is that during Customizing the user only works with those configuration activities belonging to the application components to be implemented. The selection of application components covers all clients and applies to:
 - the entire enterprise (-> Enterprise IMG)
 - individual project phases (-> Project IMG)

- ❑ In Release 3.0, the Enterprise IMG will reflect the functionality required for the installation. SAP plans to be able to make several Enterprise IMGs available in the course of extending the Business Engineering Workbench, in order to achieve a higher degree of flexibility at the enterprise level. It will then be possible to reproduce models for a subset of the clients involved, and also for individual clients in extreme cases.

Competing requirements of various clients:

- ❑ At the enterprise level, select the total set of all components which are to be implemented and used by all your customers.

- ❑ Then for each client generate one or more Customizing projects (depending on implementation strategy) and Project IMGs. Using these projects, you then perform Customizing for each client. Authorizations enable you to specify who is allowed to store status information and project documentation for each project.