

Mobile Client Technology

Release 4.0

Release Notes

Last updated: January 2004



Copyright

No part of this document may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

ORACLE® is a registered trademark of ORACLE Corporation, California, USA.

INFORMIX®-OnLine for SAP is a registered trademark of Informix Software Incorporated.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of The Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Laboratory for Computer Science NE43-358, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139.

JAVA® is a registered trademark of Sun Microsystems, Inc. , 901 San Antonio Road, Palo Alto, CA 94303 USA.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, mySAP.com, mySAP.com Marketplace, mySAP.com Workplace, mySAP.com Business Scenarios, mySAP.com Application Hosting, WebFlow, R/2, R/3, RIVA, ABAP, SAP Business Workflow, SAP EarlyWatch, SAP ArchiveLink, BAPI, SAPPHIRE, Management Cockpit, SEM, are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Contents

Mobile Application Studio (MAS)	4
Business Library (New)	4
Scriptable Business Queries (New)	4
Collision Resolution in Same Change list (New)	4
Hyperlink Wizard (New)	4
Bluebox Control Binding (New)	4
Anchor Modeling (Enhanced)	5
User Management (Enhanced)	5
Special Character Check in Code Window (Enhanced)	5
BO Delete Event Signature (Changed)	5
Import Code from VBA IDE (New)	5
Worklist and Favorites (New)	5
Relationship Modeler (Changed)	6
Facility to start MAS from Mobile Client Application (New)	6
Navigation to Underlying Objects (New)	6
Translation Tool (New)	6
Custom Resources (New)	6
UI References (Changed)	6
Slim Toolbox (Changed)	7
Copy BOL Objects (New)	7
Change List Comments (Enhanced)	7
NUMC / Currency and data type (Changed)	7
Terminology (Changed)	7
Combo Controller Modeling (New)	8
Version tree for all objects (New)	8
Main Product Executable (Changed)	8
View Output Window (New)	8
Debug Setting (New)	8
Message Editor Generation (New)	8
Supply methods for BO Properties not owned (Changed)	8
Help Modeler in MAS (New)	8
BDoc Consistency Checker (Enhanced)	9
Generator (Enhanced)	9
Incremental Generation (New)	9
Generation List (New)	9
Generation Settings (New)	9

Application Repository Services	10
SAP MAR Central Service (New).....	10
Incremental Generation (New).....	10
Better Collision Resolution (New)	10
Removal of Main Branch Changeable Collision (New).....	10
Additional User Profiles (New)	10
Mass Change of User Profiles (Enhanced)	10
Branch Name Support (New).....	10
Long Text Uniqueness for BO Property (New)	11
Loading Multiple Objects (New).....	11
Baseline Comparison Tool (New)	11
Application Tile Set Help Support (New)	11
Transport Management (Enhanced).....	11
Repository Landscape Definition Mechanism (New)	11
Transport Configurator (New).....	11
Mobile Upgrade Console (Enhanced).....	12
Upgrade Units (New).....	12
Packaging Technology (New)	12
Deployment Tool (Enhanced).....	12
Built-in Commands (New)	12
Transport Order Service (Deleted)	12
Known Issues.....	13
Business Object Layer.....	14
Support for NUMC domain (New)	14
CURRENCY domain support (New)	14
Login/Logout mechanism (Enhanced)	14
BOL model changes (Enhanced).....	14
Registry Restructuring for BOL (Changed)	14
Log messages from the application code (New)	15
Error handling enhancements (New)	15
Business Object (Enhanced)	15
Updating an existing record when the Primary Key is empty (New).....	15
Set the property of an attribute inside <i>AttributeChanged</i> event (New).....	15
<i>SaveType Property</i> in save method (Enhanced)	16
<i>Revert</i> a new business object (Enhanced).....	16
<i>BeforeDelete</i> event signature (Changed).....	16
Child business object delete (Enhanced).....	16

Marking child business objects for deletion (Enhanced)	16
Clone the entire hierarchy of a Business Object (Enhanced)	16
Business Collection (Enhanced)	17
Filter feature (Enhanced).....	17
Accessor method (New)	17
Find method for Boolean values (Enhanced).....	17
Scripting a Business Query (New).....	17
Combo Engine Refresh (New).....	17
BOL Generator (Enhanced)	17
Adaptation to Incremental Generation (Enhanced).....	18
Adaptation to the Unified Generator Framework (New).....	18
Known Issues.....	18
Calendar Control.....	19
Settings	19
Font, Time, Date and Color (New)	19
DatePicker (New)	19
Language (Changed)	19
Personalization.....	19
Personalized Settings and Status (New).....	19
Calendar View.....	20
Workweek View (New)	20
Day Format Caption in week View (New)	20
Week number of Workweek Template View (New).....	20
Accessibility (New).....	20
Appointments (Enhanced)	20
Start/End Day (New)	20
Additional Time Zone (New)	20
Interface Name (Changed)	20
OnPersonalization Event (Changed)	21

The release notes for Mobile Technology release 4.0 cover the following scope:

- Mobile Application Studio
- Application Repository Services
- Business Object Layer
- Calendar Control

Mobile Application Studio (MAS)

Note: For more detailed information about the new and enhanced features listed below, see the online help for *Mobile Application Studio* release 4.0 available in the help portal.

Business Library (New)

All Business Object Layer (BOL) objects are now grouped together as a business library. The immediate benefit is the possibility of indicating a user specific name for the generated file `sfabol.dll`. Furthermore, the application developer can add references to the BOL project in MAS instead of making changes in the `BOLREF.ini` file. The introduction of the business library brings in an additional hierarchy in the *Object Browser*.

Scriptable Business Queries (New)

All business queries are now scriptable. You can add logic to the business query by scripting the business query. This is done by including a condition for more appropriate and controlled querying. All scriptable business queries can handle *BeforeQuery*, *OnQuery* and *AfterQuery* events. Scripting a business query adds technical value because it allows you to define further business.

Collision Resolution in Same Change list (New)

You can now resolve collisions in the change list where the collisions occurred. Earlier you had to create a new change list to resolve collisions that have been caused by an existing change list.

Hyperlink Wizard (New)

Hyperlink modeling has been simplified at the tile and application level. Earlier you were required to navigate between the tile and application to create hyperlinks. Now you can create source and target hyperlinks in a single procedure.

Furthermore, you had to manually create hyperlinks when copying business components. With this release, you can indicate whether the hyperlinks have to be imported when a business component is copied. Additionally, you also have a visual representation of all hyperlinks within a tile.

There is a new type of hyperlink called 'OnNew'. This provides application users with an additional menu option to create new objects in the tile that defines the hyperlink. For example, in the *Mobile Sales* application, during runtime, if a user selects 'New' from the *Contacts* tile set, the corresponding detail tile set for *BOActContact* starts.

Bluebox Control Binding (New)

The bluebox control can now be bound to different controls or business object (BO) attributes at design time. At runtime, the bluebox responds to the corresponding attribute change events.

Anchor Modeling (Enhanced)

The *Anchor Modeling* screen (formerly known as the Anchor Modeling Wizard) has been enhanced to simplify the creation of anchors, anchor mappings and anchor hierarchies for tile sets and business components. This screen now provides a visual representation of the anchors, by listing them in the tiles or business components (whatever is the case), making it easy to use.

User Management (Enhanced)

Additional user profiles are provided for MAS users for different purposes. Users can now logon to MAS using profiles such as administrator, developer, information developer, quality responsible or guest. The tasks permissible to users based on their profiles include creating and maintaining other user records, maintaining development objects in the Mobile Application Repository (MAR), releasing change lists, modifying passwords and framework objects, integrating help topics for interface objects, transfer ownership of objects and much more.

Special Character Check in Code Window (Enhanced)

MAS users are now prompted with a message whenever code is written or pasted into the current application code. This message is related to a check that is made for the presence of special characters or keywords in the code. Earlier there was no check made if special characters were inserted in to the code and this generated errors during compilation. Now users are informed when special characters are pasted into the code and they can choose to save the code with these changes.

BO Delete Event Signature (Changed)

The BO delete event signature has been changed to *BeforeDelete* (*byref cancel as Boolean*) from *BeforeDelete()*.

For more information, see the BOL section in this document.

Import Code from VBA IDE (New)

A new service, sync code, is introduced to directly import changes, made in the Integrated Development Environment (IDE), into the MAR. This is possible when the IDE is opened outside MAS. Changes must be saved after the import. However, if the IDE is started from within MAS, saving changes is not necessary. The sync code service enables MAS to compare the contents of the MAR and the latest Visual Basic for Applications (VBA) project. The result displays a list of code units that differ. The MAS user can then choose whether or not to import the changes into the MAR. Objects that can be imported are:

- BOL related - methods, event handlers, user exits, supply methods, rules
- UI related - custom methods, event handlers, event signatures, free scripts and supply functions

Worklist and Favorites (New)

The new tab, *My Objects*, available in the *Object Browser* provides the MAS users with the option of working with a *worklist* and *favorites*. A *worklist* is list of objects contained in the current active change list. All operations supported on objects in the *Object browser* are also supported in a *worklist*. Objects such as short text, documentation objects and custom resources are not displayed in a *worklist*.

Favorites allow you to add shortcuts to the frequently used development objects to its favorites list. Furthermore, it allows you to organize these shortcuts and related folders.

Relationship Modeler (Changed)

The relationship modeler has been redesigned to allow MAS users to group related information.

Facility to start MAS from Mobile Client Application (New)

MAS users can now access MAS directly from the runtime environment of a mobile client application. This feature benefits users by providing a dynamic environment of accessing the development platform while testing the runtime application. From a specific screen in the mobile application, developers can open the corresponding design screen of a tile, tile set, business component or application in MAS.

Navigation to Underlying Objects (New)

MAS users can now navigate to an underlying object from any other object. Navigations supported are:

- Combo engine from a business object (BO) or business query (BQ) property
- Contained BO from a BQ
- Source and target BO from a relation
- BO or BQ from any anchor
- BO or BQ property from any control in a tile

Translation Tool (New)

An external translation tool has been implemented. This replaces the Multilingual Translator that existed in previous releases. It provides features to translate text objects, to export text to other systems as well as to import text from other systems (typically R/3 systems) using XML.

Note: For more detailed information, see the online help for *Translation Tool* release 4.0 available in the help portal (<http://help.sap.com>).

Custom Resources (New)

Every mobile client application can now have its own logo and splash image attached. Furthermore, users can add special images (for example, icons) on buttons in a tile. Custom resources can be attached for all interaction components. These resources include file types such as .jpg, .txt and so on, and are stored in the MAR.

UI References (Changed)

The reference grid for all interaction components has been removed. A new screen has been introduced that is similar to Visual Basic (VB) *Add references* screen. This screen displays all types of libraries that are available on the system and allows the users to choose the required libraries.

Slim Toolbox (Changed)

The toolbox has been trimmed down to a thin strip and now appears next to the *Object Browser*. This has been changed to provide more working space for the *Property* window.

Copy BOL Objects (New)

MAS users can now copy all BOL objects (business objects, business queries, combo engines and rules) along with their child objects. This option was not available earlier.

Change List Comments (Enhanced)

MAS users must enter comments for releasing a change list.

NUMC / Currency and data type (Changed)

There are two new data types that have been introduced - NUMC and Currency for BO properties and user interface controls.

For more information, see the BOL section in this document.

Terminology (Changed)

The terms (interface text) that have been changed in the current release are listed below.

From	To
changelist	- change list
relation	- relationship
controller	- control
tileset	- tile set
choice list/choice field	- list box
user/master data	- business data
script object	- code object
UI elements	- screen elements
ARS / ARS DB / AR	- Mobile Application Repository (MAR)
IDES	- user database
Transaction Layer	- BDoc layer
replication	- synchronization
client upgrade source	- mobile upgrade console
tile container	- tile
tile set container	- tile set
tile set window	- tile set design area

Combo Controller Modeling (New)

MAS Users can now create any number of columns for a combo control in a tile. When BDoc validation is ON, new columns can be added and mapped to the underlying segment fields of the combo engine.

Version tree for all objects (New)

MAS users can now directly view the version tree for all objects of any open form. This option is available via the menu bar. Earlier you could only see the version tree from the object in the object browser.

Main Product Executable (Changed)

As per product standards the main workbench executable has been renamed from `workbench.exe` to `SAP_MAS.exe`. You are required to change the necessary shortcuts accordingly.

View Output Window (New)

The output window can be opened from the menu to show the last generation results.

Debug Setting (New)

It is now possible to choose only BOL or a UI project to debug.

Message Editor Generation (New)

Messages can now be generated directly from the *Message Editor* screen.

Supply methods for BO Properties not owned (Changed)

The restriction of not being able to create supply methods for BO properties that are not owned, has now been removed. Supply methods improve performance at runtime. Checking the ownership of business properties has been removed, allowing customers to create supply methods for BO properties created and provided by SAP.

Help Modeler in MAS (New)

Application users now have the option of viewing help specific to a tile set (context sensitive help) in the mobile client application. Earlier, users were provided with the entire application help (displaying the first topic) from the Help menu.

A new tool, *Help Modeler* is implemented to enable the MAS users to integrate tile set specific help with the mobile client applications. It can be done by indicating a help context ID for a selected tile set. This ID is a Logical Information Object (LOIO) created in Knowledge Warehouse (SAP authoring tool). The *Help Modeler* also allows you to test the integrated link by indicating the required Help filename (CHM). When using the *Help Modeler* you can filter the list of tile sets that do not have context IDs indicated.

BDoc Consistency Checker (Enhanced)

The consistency check for BDocs is much faster now with the use of the *Prefetch* mechanism. Furthermore, this feature now has a progress indicator and a button for stopping the check at any time.

Generator (Enhanced)

Incremental Generation (New)

An *Incremental Generation* has been introduced to track and store valid runtime objects that are generated by the BOL and UI framework generators. This allows you to generate runtime objects on demand when development objects are modified. Moreover, this prevents unnecessary generation of unmodified development objects.

Generation List (New)

This feature enables the MAS users to add objects that require generation, to a generation list. This list can be saved and the generation can take place at a later time.

Generation Settings (New)

The *Generator* has been further enhanced with a new feature for defining settings and storing these settings in a file that can be reused later. Generation settings can include defining target paths and storing this information in a particular file. Similarly, different generation settings can be grouped and stored in separate files for generation at a later time or reuse.

Application Repository Services

SAP MAR Central Service (New)

With the 4.0 release, a new component, SAP MAR Central Service, has been implemented to replace the Short Term Lock Server (STLS), existing in the previous release. This Central Service is a Java-based component that supports the Lock Manager (for locking support) and Validation Manager (for incremental generation support). This component is installed as a service during the installation of the mobile application repository.

Incremental Generation (New)

This feature has been implemented for the User Interface and BOL generators. This has been done to increase the speed of the generation process for Runtime Objects (RTOs). Incremental generation also facilitates the reuse of Runtime Objects already generated, across different Mobile Development Workstations. This feature ensures that RTOs are not generated (instead, reused) for objects that are not modified.

Better Collision Resolution (New)

This feature has been implemented to resolve collisions in the same open change list that caused the collision. Now you do not have to create a third change list to resolve a collision.

Removal of Main Branch Changeable Collision (New)

This feature has been implemented to avoid reporting the main branch changeable collision. This is because the main branch changeable collision is always resolved on the main branch version. This causes the following changes:

- Baseline is allowed to reside on a side branch even if the object is owned in the current Mobile Application Repository (MAR)
- During modification of a baseline, which is on a side branch, an implicit MergeCopy is performed with the last main branch version

Additional User Profiles (New)

This feature has been implemented to introduce controlled access to the MAR. Some of the highlights of this feature are listed below:

- Change list release during correction and emergency correction phases is controlled.
- Users of the framework profile only can modify the framework objects.
- Users of the Info developer profile can access the text objects even after the UI freeze.

Mass Change of User Profiles (Enhanced)

It is now possible to change the profiles of multiple users. This is useful when access rights have to be changed for multiple users during different development phases.

Branch Name Support (New)

This feature has been implemented to allow a consistent usage of branches. A list of predefined branches with meaningful branch names can be stored with MAR. Later, during development, these branch names must be used.

Long Text Uniqueness for BO Property (New)

This feature has been implemented to ensure uniqueness of the long text of a business object property. This restricts you from entering a duplicate long text for a property of the same business object. This validation occurs before generation of the business object, which was not the case in the previous release.

Loading Multiple Objects (New)

This feature has been implemented to speed up the loading of related objects along with the main object for a specified navigation path.

Baseline Comparison Tool (New)

This tool is used to compare the baselines and change lists between two MARs. This helps you check whether the MARs are in sync or a transport is required. This tool allows you to view the:

- Baseline objects of source and target MARs
- Objects with different baselines in source and target MARs
- Objects that exist only in one MAR
- Change lists that exist in one MAR

Application Tile Set Help Support (New)

This feature has been implemented to support the integration of context-sensitive help with a mobile client application. The info developer profile allows this integration using the *Mobile Application Studio* (MAS). This feature causes the following changes:

- The automatic collision detection is not provided for these objects.
- The UI generators extracts the information regarding the integration of tile set help so that it is available in the runtime repository.

Transport Management (Enhanced)

Repository Landscape Definition Mechanism (New)

This new mechanism has been introduced to allow you to automate the replication or transport processes in an extended repository landscape. This has been provided for the purpose of replication or transport of repository data. Furthermore, the transport or replication processes are now automated within this extended landscape. This mechanism potentially reduces the manual efforts required for performing these processes.

A new transaction /NSMOBREPMAIN has been implemented to register MAR IDs in the *CRM Server*.

Transport Configurator (New)

The Transport Configurator has been implemented to replace the 'Transport Installer'. The configuration allows you to install the ARS Transport Agent (TA service) on the Mobile Repository Server (MRS). Additionally, you can create new destinations to facilitate communication between the MRS and the *CRM Server*. When the TA service is started, it uses the existing destinations. This speeds up the start of the TA service, since no destination needs to be created.

The transport configuration tests the connection between the MRS and *CRM Server* as soon as there is a connection. Earlier this was not possible with the 'Transport Installer'. You can also use the transport configuration, to change user and password information used by the TA service for logging into the MAR.

Mobile Upgrade Console (Enhanced)

Upgrade Units and Standard Units (New)

In the current release, CRM 4.0, upgrades comprise one or more *Upgrade Units*. A unit may contain files, folders and commands. In earlier versions, an upgrade directly consisted of files, folders and commands. *Upgrade Units* bring in more flexibility when building upgrades. You can create specific *Upgrade Units* and later include these *Upgrade Units* in *Upgrade Scripts*.

SAP ships Standard Units that customers can use to deploy standard upgrades. For example, standard upgrades apply to upgrading the User Interface, Business logic and BDocs of the application. Additionally, customers can create their own Upgrade Units, containing a specific customization, and include these units in an Upgrade Script.

Working Modes (New)

Mobile Upgrade Console allows you to work in two modes. The Unit mode allows you to create / modify / delete Upgrade Units. The Upgrade mode allows you to add/remove Upgrade Units to/from an Upgrade Script, and set properties of the upgrade.

Packaging Technology (New)

This feature enables the Mobile Upgrade Console to create a single package for an upgrade with the `.mup` file extension ('Mobile Upgrade Package'). The structure of this package will allow you to view the entire contents of the package. Users can also delete units from a package or change the properties of a package prior to deployment.

Viewable Upgrade Package (New)

You can view the contents of a *Mobile Upgrade Package*, on mobile clients, using the *Mobile Upgrade Deployer*. The contents display a list of *Upgrade Units* with contents. Furthermore, at the time of deployment you can indicate specific *Upgrade Units* that you want deployed. This feature brings in the flexibility of indicating contents of a *Mobile Upgrade Package* prior to deployment. You can also set a property to indicate if the *Mobile Upgrade Package* is modifiable by the application user.

Deployment Tool (Enhanced)

The enhanced Mobile Upgrade Deployment tool allows you to view and modify the contents of the `.mup` file before starting deployment. It also provides you detailed information about the status of the execution of the upgrade.

Built-in Commands (New)

New built-in commands have been implemented to perform common tasks while upgrading the mobile clients. These common tasks include copying, moving or deleting files and folders, running batch files and executing programs.

Transport Order Service (Deleted)

Earlier a *Microsoft Windows NT* service used to provide MAS with Remote Function Call (RFC) access to the *Change and Transport System* (CTS). In order to decrease the installation and

administration overhead, this service ('Transport Order Service') has been removed completely and its functionality has been directly integrated with MAS.

Known Issues

- `StorageService.dll` and `MessageStore.ars` must be in the same directory.
- In the `MessageStore.ars` file an error or message description must not end with a parameter. If you want to place a parameter at the end of the description, leave an empty space between the parameter and description.

Business Object Layer

NUMC Data Type (New)

Prior to this release, there were different methods in the application code to define a string to store data in a special length with leading zeros. With the current release, the NUMC data type has been introduced to accomplish this for attributes of a business object (BO) or business query (BQ). This data type stores the data of the attribute as a string value in the application database. It is similar to the NUMC data type that exists in the *CRM Server*. Therefore, this data type permits a better exchange of data between mobile clients and the *CRM Server*. The Business framework considers the NUMC data type as a "string" data type.

CURRENCY Data Type (New)

With the current release, the Currency data type is introduced for attributes of a business object (BO) or business query (BQ). This data type displays currency dependent values with decimal places based on the currency. For example, YEN is displayed without decimals, where as USD is displayed with two decimals. Business framework considers the Currency data type as a "double" data type.

Login/Logout mechanism (Enhanced)

Prior to this release, the business logic for the login mechanism was built into the user interface. With the current release, the business framework provides a Login/Logout mechanism at the framework level. There were a number of issues that have been cleared with this new mechanism. First, the concept of logging into the mobile applications did not follow a definite flow. Furthermore, the Login process earlier required plugging in business logic at the user interface, thus permitting virtually any written code to be included the login events. As a result of the new mechanism, the Login process has been moved to the Business Object Layer.

BOL model changes (Enhanced)

The meta model has been enhanced to enable an application developer to:

- Specify external library references for a particular business library
- Validate a user as part of the login process

Registry Restructuring for BOL (Changed)

All BOL related registry keys have been moved to a single location in the registry.

`HKEY_LOCAL_MACHINE\SOFTWARE\SAP\CRM\Mobile\BOL`

BOL will not access any registry key in these locations:

`KEY_LOCAL_MACHINE\SOFTWARE\SAP\CRM\BOL`
`HKEY_LOCAL_MACHINE\SOFTWARE\SAP\MSA`

However, generators read a registry entry "RootDir" in the following location to get the root installation path of Mobile Sales. This is needed by generators to generate a relative path for RTOs, that are used by business framework at runtime, in the corresponding `.reg` file generated by each generator.

`HKEY_LOCAL_MACHINE\SOFTWARE\SAP\MSA`

- Generators (BOL VBA Generator, RT Generator, MsgInfo Generator) read the registry entry "RootDir" under HKEY_LOCAL_MACHINE\SOFTWARE\SAP\MSA to get the MSA installation path.
- If the root installation path is a substring of the generation path provided to the generators, then the relative path (represented by the registry entry "RelativePath") is generated in the respective .reg files and the absolute path (represented by the registry entry "Path") is set to blank.

Log messages from the application code (New)

With the 4.0 release, the Business Object Layer (BOL) has introduced a feature that logs messages originating from the application code and that were not required to be displayed to the end user. Earlier, messages (errors/warnings/information) in the framework or application code were shown to the end user in the user interface (status bar or window).

Once the messages are logged, the application developer can later reference this log for diagnostic purposes. This is especially beneficial when there is a need to log debug messages. However, should the end user choose to view logged messages, this can be done using the Analyzer tool available in the Mobile Sales application. The Analyzer tool requires that `UFTTools.exe` be registered on the machine.

Furthermore, the application developer can programmatically enable/disable logging of messages. A new method, *LogMessage* (message as String, Optional logpriority as *BILogPriorityType*, Optional source as object, Optional instanceName as String, Optional subroutine as String), has been provided for the *BusinessRootObject* interface to log messages.

Error handling enhancements (New)

In this release, the error component provides the application developer with a framework where he can categorize internal error messages as *EMS_ERROR*, *EMS_ERROR_HIGH*, *EMS_WARNING*, *EMS_WARNING_HIGH*, *EMS_INFO* and *EMS_INFO_HIGH*. Existing error types such as *Information*, *Critical*, *User Error* or *Warning* have been marked deprecated and should not be used for new development. However, keeping in mind backward compatibility the framework still supports these error types. Thus enabling you to raise messages from the application code, classify it as error, warning or information and associate the message with the control in the user interface.

Business Object (Enhanced)

Updating an existing record when the Primary Key is empty (New)

With the 4.0 release, the business framework performs a check while creating a new business object (BO). Prior to this release, records that had empty primary keys were created in the user database and consolidated database (CDB).

Now, the check prevents records with empty primary keys from being created. This check is valid only in cases where the SQL triggered for creating a new BO has a WHERE clause as PRIMARY KEY = ''.

Set the property of an attribute inside *AttributeChanged* event (New)

With the 4.0 release, the business framework allows you to change the property for an attribute inside the *AttributeChanged* event, of the same attribute. For example, you can change the

property of the attribute *blAttributeNormal* to *blAttributeMandatory*). This was not possible in previous releases.

However, the attribute property cannot be changed for an attribute within the *AttributePropertyChanged* event of the same attribute. This is because it would lead to an infinite loop (changing the attribute property inside the *AttributePropertyChanged* event and running the *AttributePropertyChanged* event again).

SaveType Property in save method (Enhanced)

SaveType – the business framework allows the application developer to set the *savetype* as LOCAL or COMMUNICATABLE by:

- Setting the business object attribute SAVETYPE to S or C
- Passing the value S or C directly to the save method

By passing the letter S, the saved business object is available locally. By passing C the saved business object is available to other users.

Revert a new business object (Enhanced)

With the current release the developer is informed whether a business object was a new business object or not in the *Deletedevent* of the business object.

BeforeDelete event signature (Changed)

The *BeforeDelete* event of a business object is changed to accept a Boolean parameter. This enables a developer to specify whether to cancel the respective operation in the event.

Child business object delete (Enhanced)

The *BeforeDelete* event has been enhanced to indicate whether the deletion of a child object is also applicable to the parent object.

Marking child business objects for deletion (Enhanced)

Prior to this release, when a child business object was deleted, a save was performed on each parent object of the deleted child business object.

With the current release, child business objects are marked for deletion. The business framework triggers an event when objects are marked for deletion. The parameter of the event indicates whether the BO is marked for deletion, unmarked for deletion programmatically or unmarked for deletion by the framework. Marked objects get deleted when the parent business object is saved or deleted or when the marked business object is saved or deleted.

Clone the entire hierarchy of a Business Object (Enhanced)

With the 4.0 release, it is possible to clone the entire hierarchy of a Business Object (BO). The signature of the existing *Clone()* method has been enhanced to allow the application developer to choose to clone the BO on which the *Clone()* method is run (Shallow Clone) or the entire hierarchy of that BO (Deep Clone). Furthermore, it is possible to specify whether you want to delete parent information of the BO on which the *Clone()* method is run. The semantics have changed with the way events are run during a clone and two new events (*BeforeChildClone ()* & *Cloned ()*) have been introduced to provide the application developer with more flexibility while cloning the hierarchy of a BO.

Prior to this release, when a BO was cloned, the *Loaded()* event for the BO was triggered. From now on, the *Loaded()* event will not be run.

The *Cloned()* event is run on a BO after the new BO is created and all its properties copied as part of cloning. The listeners of this event can overwrite the attribute values if required. However, if attributes are changed in the event handler, then the *AttributeChanged* event will be run.

The *BeforeChildClone()* event is run before a relation of the BO is cloned. It is also *cancel the cloning of a relation.

Business Collection (Enhanced)

For improved performance of the business object layer, enhancements have been made to two existing methods and one new method has been added to the business collection. These methods optimize response time and other resources such as memory usage.

Filter feature (Enhanced)

Prior to this release whenever the *Filter* method was called the business object layer used to fetch the underlying *recordset* of the business collection and filter this newly fetched *recordset*. This appeared to be a performance bottleneck, which resulted in lesser usage of the function. With the 4.0 release, these issues have been addressed. The *Filter* method is applied before fetching the underlying *recordset*. This results in a performance gain.

Accessor method (New)

The Accessor method allows direct access to a business object that is in a dirty state or created through a business collection. This method eliminates the task of looping through the entire hierarchy of a business collection.

Find method for Boolean values (Enhanced)

The *Find* method in a business collection has been enhanced. It is now possible to perform a search on a Boolean field with the search criterion, `<true>` or `<TRUE>`.

Scripting a Business Query (New)

Scripting a Business Query allows you to enhance the logic of a business query. You can define your own SQL query in the application code by referring to an existing business query. This feature adds technical value because it allows you to define further business logic for a business query. Most importantly, this new feature has been introduced to avoid adding business logic in the UI framework, as was the case earlier.

Combo Engine Refresh (New)

A new feature *Combo Engine Refresh* has been introduced with the current release. This feature enables a combo list to be updated during runtime. Earlier, it was not possible to refresh combo data programmatically. This refresh depended on combo access. Now the developer can refresh the combo list programmatically whenever needed, allowing more flexibility while writing business logic.

BOL Generator (Enhanced)

With the current release, the BOL Generator has been enhanced for better performance and optimization of resources such as memory usage.

Adaptation to Incremental Generation (Enhanced)

The *Adaptation to Incremental Generation* feature enhances the performance of the generator. The generator can now perform generation only if there are modifications subsequent to the previous generation.

Adaptation to the Unified Generator Framework (New)

Adapting to the *Unified Generator Framework* enables a border between generators of the framework and generators of the *CRM Mobile Application Studio*. A new template based enhanced generator is available to use the Generator framework. The Unified Generator has the following features:

- A common and easy-to-use user interface across all generators
- A tree based view for selection of objects rather than a list
- Extended logging of messages (errors/warnings/information) during generation
- A common format for log files generated across all generators
- Control over output and storage of generated log files
- Incremental generation of Runtime Objects
- Validation of objects during generation

Known Issues

- A query on the NUMC attribute with a search criteria "<value>*" is not supported and is currently a limitation in the framework.
- When *Refresh on Combo Engine* is called the Combo Engine data is refreshed and not the business object (BO) attribute associated with the Combo Engine. Therefore, if the Combo Engine data changes as part of *Refresh*, there could be a possibility where the BO attribute still points to the old value. The application developer needs to explicitly call the 'MapShortToLong' or 'MapLongToShort' method after a *Combo Refresh* to refresh the BO attribute data.
- When the Visual Basic for Applications (VBA) project is opened in the VBA Integrated Development Environment (IDE) and changes made to the project, the project becomes dirty. So even if no modifications have been made to the project, the VBA IDE prompts the user to save changes, when the project is closed. This has been a general limitation in VBA IDE. This has now been solved. However, if you want to import code changes, made in the VBA IDE, into the repository, you need to save changes made and then close the VBA IDE. Currently the limitation is that if you choose to close the VBA IDE without trying to save changes first, even though you will be prompted to save the changes, the saved changes will not imported into the repository.

Calendar Control

Settings

Font, Time, Date and Color (New)

With the 4.0 release, application developers can set the following details for the calendar:

- Font size (property *CalendarFontSize*)
- Font name or font family calendar (property *CalendarFontName*)
- Time format (property *TimeFormat*)
- Short date format (property *ShortDateFormat*)
- Long date format (property *LongDateFormat*)
- Number of days for the work week view (function *SetDaysOnWorkWeek*) - default days for a work week are 5
- Date format for the day view caption (method *SetDateFormat*)
- Color for working and non-working time, when viewing appointments (property *WorkingTimeColor*, *NonWorkingTimeColor*)
- Type of work week view
- Additional time zone (method *SetAdditionalTimeZone*)

DatePicker (New)

The *Sheridan* type DatePicker has been introduced and month wise access of the calendar is provided. New properties have been introduced to enable the user to associate or get an associated style with the following DatePicker styles:

- DatePicker footer style (property *DatePickerFooterStyle*)
- DatePicker week number style (property *DatePickerWeekNumberStyle*)
- DatePicker week header style (property *DatePickerWeekHeaderStyle*)

Language (Changed)

With the 4.0 release, a new property has been introduced that will enable the users to associate or get the language set for the calendar control. This is a read/write property and has been changed for maintaining naming conventions of properties of SAP standards (property *Language*). Earlier this property was called 'Locale_R/3'.

Personalization

Users can personalize the appearance of the calendar for all sessions. This is an industry standard user friendly feature. A user can personalize the appearance of the calendar by defining specific colors for working and non-working times and time zones.

Personalized Settings and Status (New)

Users can be authorized to view the personalization settings screen (property *EnablePersonalization*). If the value of the property is set to 'True', the user can access the personalization screen from the Context menu.

Alternately, there is a method that enables the application code to display the personalization window from the back end instead of using the Context menu (method *ShowPersonalizationPopUp*).

Users can set the personalization field status for the calendar (function *SetPersonalizationFieldStatus*).

Once settings are made, users can view the status of all personalized settings for the calendar. This function provides information about what fields can be personalized and whether the current status is 'enabled' or 'disabled' (function *GetPersonalizationFieldStatus*).

Calendar View

Workweek View (New)

This is an additional view that displays multiple days as columns in a single row. It is a subview of the appointment view.

Day Format Caption in week View (New)

Users can get information on the format of the date for the day view caption (method *GetDateFormat*).

Week number of Workweek Template View (New)

This method allows the application developer to know the reference week number of a Workweek view, when the application user navigates to either the next or previous Workweek. When the calendar is displayed in the Workweek view for the first time the reference week number is considered as the week number zero (method *GetWeekNumberForWorkWeekTemplate*).

Accessibility (New)

The calendar control is accessibility compliant. It can be used by visually challenged people. The application developer can enable or disable the accessibility feature for the calendar control (property *EnableAccessibility*).

Appointments (Enhanced)

With the 4.0 release, application users can create an appointment that spans across a number of days as well as on a single day basis (method *CreateAppointmentByDay*).

Start/End Day (New)

With the 4.0 release, two functions have been introduced to return values for selected start and end days (functions *GetSelectedStartDay* and *GetSelectedEndDay*).

Additional Time Zone (New)

This function returns Day Light status and an additional time zone set for the calendar (method *GetAdditionalTimeZone*). The additional time zone is set using the *SetAdditionalTimeZone* method. For more information, see the section Settings described earlier within Calendar Control.

Interface Name (Changed)

There has been a change incorporated in the interface name to maintain SAP Naming Standards.

- Previous name: *myCalendar*
- New Interface name: *Calendar*
- Interface name recorded in the library: *SAPCalendar*

OnPersonalization Event (Changed)

There has been a signature change of the *OnPersonalization* event to notify the application code about the changes that have taken place in the personalization settings of the calendar.