



## **Progress Dynamics. Product Update Bulletin**

© 2005 Progress Software Corporation. All rights reserved.

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document.

The references in this manual to specific platforms supported are subject to change.

A [Stylized], Allegrix, Allegrix & Design, Business Empowerment, eXcelon, ObjectStore, PeerDirect, Progress, Powered by Progress, Empowerment Center, Progress Empowerment Center, Progress Empowerment Program, Progress Fast Track, Progress OpenEdge, Progress Profiles, Partners in Progress, Partners en Progress, Progress en Partners, Progress in Progress, P.I.P., Progress Results, Progress Software Developers Network, ProVision, ProCare, ProtoSpeed, SmartBeans, SpeedScript, Technical Empowerment, and WebSpeed are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, A Data Center of Your Very Own, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Cache-Forward, Fathom, Future Proof, IntelliStream, ObjectCache, ObjectStore Event Engine, ObjectStore RFID Accelerator, ObjectStore Trading Accelerator, OpenEdge, POSSE, POSSENET, ProDataSet, Progress Business Empowerment, Progress for Partners, PSE Pro, PS Select, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Any other trademarks and service marks contained herein are the property of their respective owners.

February 2005



Product Code: 3857  
Item Number: 101926;V2.1B

# Contents

---

<b>Preface</b>	<b>vii</b>
Preface	vii
Audience	vii
Organization of this manual	vii
Typographical conventions	viii
<b>1. Overview of New and Revised Features</b>	<b>1-1</b>
1.1 Extending object classes	1-2
1.2 New APIs for dynamic lookups and combos	1-2
1.2.1 keep_old_API session parameter	1-2
1.2.2 New LookupField class	1-2
1.2.3 New and old API comparison	1-3
1.3 New application data caching on the client side	1-4
1.4 Manager API changes	1-5
1.5 Repository schema changes	1-5
1.6 ADM2 changes	1-5
1.7 Web application development	1-6
1.8 Translation	1-6
1.9 Deployment	1-7
1.9.1 Batch-mode Dynamics Configuration Utility	1-7
1.9.2 Site Data Dump and Load utilities	1-8
1.9.3 Multi-transaction sequence deployment	1-8
1.9.4 Static-4GL equivalents of dynamic objects	1-8
1.9.5 Deployment between versions	1-9
1.10 Print Preview	1-9
1.11 Documentation set changes	1-10

<b>2.</b>	<b>Feature Comparisons</b>	<b>2-1</b>
2.1	DCU on run-time client	2-2
2.2	Multi-transaction sequence deployment	2-2
2.3	Clarification of Save Dynamic Object as Static menu command	2-2
2.4	Extending object classes	2-2
2.5	Manager API changes	2-3
2.5.1	Public and private APIs	2-3
2.5.2	Configuration File Manager	2-3
2.5.3	Connection and Service Type Managers	2-4
2.5.4	Customization Manager	2-5
2.5.5	General Manager	2-5
2.5.6	Profile Manager	2-5
2.5.7	Referential Integrity Manager	2-6
2.5.8	Repository Managers	2-6
2.5.9	Security Manager	2-9
2.5.10	Session Manager	2-10
2.5.11	User Interface Manager	2-11
2.5.12	Web Request Manager	2-12
2.6	Repository schema changes	2-12
2.6.1	New tables	2-12
2.6.2	New fields	2-12
2.6.3	Index changes	2-12
2.7	ADM2 changes	2-13

---

Table 1–1:	Old field API to new field API map . . . . .	1–3
Table 1–2:	Changes to documentation in Version 2.1B . . . . .	1–10
Table 2–1:	Changes to Profile Manager API calls . . . . .	2–5
Table 2–2:	Changes to Repository Manager API calls . . . . .	2–7
Table 2–3:	Changes to Repository Design Manager API calls . . . . .	2–9
Table 2–4:	Changes to Session Manager API calls . . . . .	2–11
Table 2–5:	Changes to Session Manager properties . . . . .	2–11
Table 2–6:	New Repository fields . . . . .	2–12
Table 2–7:	Changes to Repository indexes . . . . .	2–12
Table 2–8:	New ADM2 API calls . . . . .	2–13
Table 2–9:	Changed ADM2 API calls . . . . .	2–14
Table 2–10:	New ADM2 properties . . . . .	2–14
Table 2–11:	Changed ADM2 properties . . . . .	2–15



# Preface

---

## Preface

This bulletin provides a brief description of the new and modified features of Progress Dynamics® Version 2.1B.

## Audience

This bulletin is written for Progress Dynamics application developers who are upgrading their license from Progress Dynamics Version 2.1A to Version 2.1B.

## Organization of this manual

This bulletin is organized as follows:

### [Chapter 1, “Overview of New and Revised Features”](#)

Summarizes the product features that are new or modified since Version 2.1A. Entries include links to further information in the full documentation set.

### [Chapter 2, “Feature Comparisons”](#)

Describes how changed features in Version 2.1B might affect application code written in Version 2.1A.

## Typographical conventions

This manual uses the following typographical conventions:

- **Bold typeface** indicates:
  - Commands or characters that the user types
  - That a word carries particular weight or emphasis
  - Names of user interface elements
- *Italic typeface* indicates:
  - Progress variable information that the user supplies
  - New terms
  - Titles of complete publications
- Monospaced typeface indicates:
  - Code examples
  - System output
  - Operating system filenames and pathnames

The following typographical conventions are used to represent keystrokes:

- Small capitals are used for Progress key functions and generic keyboard keys.  
**END-ERROR, GET, GO**  
**ALT, CTRL, SPACEBAR, TAB**
- When you have to press a combination of keys, they are joined by a hyphen. You press and hold down the first key, then press the second key.  
**CTRL-X**
- When you have to press and release one key, then press another key, the key names are separated with a space.  
**ESCAPE H**  
**ESCAPE CURSOR-LEFT**



---

## Overview of New and Revised Features

This chapter summarizes the product features that are new or modified since Version 2.1A. Each of the following sections includes a reference to further information in the Progress Dynamics® documentation set:

- [Extending object classes](#)
- [New APIs for dynamic lookups and combos](#)
- [New application data caching on the client side](#)
- [Manager API changes](#)
- [Repository schema changes](#)
- [ADM2 changes](#)
- [Web application development](#)
- [Translation](#)
- [Deployment](#)
- [Print Preview](#)
- [Documentation set changes](#)

## 1.1 Extending object classes

In Version 2.1B, the framework provides new capabilities for extending object classes. Most importantly, you can now create custom classes off the middle of the class hierarchy, and not just off the bottom.

<b>Manuals:</b> <a href="#"><i>Progress Dynamics Programming Handbook</i></a>
---

## 1.2 New APIs for dynamic lookups and combos

To complement performance and application data caching features in Version 2.1, the class structure and the ADM2 APIs for dynamic lookups and combos have changed.

### 1.2.1 keep\_old\_API session parameter

Progress Dynamics uses the new field API by default. If you currently have code customizing the behavior of lookups and combos, then you might want to continue using the old API. You might also want to use the old API until such time as you have thoroughly tested your applications with the new API enable. In either case, whether you need to update code or test your applications, Progress Software Corporation strongly recommends that you switch to the new API as quickly as possible. The new API is a thinner client with performance benefits. The old API will not be available in future versions of Dynamics.

Dynamics provides a session parameter to control which version of the API should be used. Set `keep_old_API` to YES if you want to continue using the old field API. Set it to NO if you want to use the new API. For each session type where you do not explicitly set this parameter to YES, Dynamics will use the new field API.

### 1.2.2 New LookupField class

To support new caching mechanisms, there is a new class, `LookupField`, in the ADM2. In previous versions, the `Field` class is the parent of the `DynCombo` and `DynLookup` classes. Now, the `Field` class is the parent of the `LookupField` class. In turn, the `LookupField` class is the parent of the `DynCombo` and `DynLookup` classes.

The `LookupField` class now contains all the properties that were common to the `DynCombo` and `DynLookup` classes. The functions for getting and setting these properties are also now a part of the `LookupField` class.

The normal way to create custom subclasses here would have been to subclass `DynCombo` or `DynLookup`. If you did this, the new class structure in itself should not affect your code. Your

use of any API calls might need to be modified, however. In the future, you should continue to subclass DynCombo and DynLookup. LookupField should not be subclassed.

### 1.2.3 New and old API comparison

[Table 1–1](#) lists the old API calls and their equivalents in the new API calls, if there is an equivalent. The Continued support? column indicates whether or not the old API call will still function when the new field API is enabled. It is likely that even these old calls will not be supported in future versions of Progress Dynamics.

**Table 1–1: Old field API to new field API map**

(1 of 2)

Old API	Continued support?	New API	Notes
getLookupQuery	Version 2.1B only	prepareField	–
getComboQuery	Version 2.1B only	prepareField	–
displayCombo	Version 2.1B only	displayField	–
displayLookup	Version 2.1B only	displayField	–
refreshChildDependancies	Version 2.1B only	notifyChildFields	Followed by call to retrieveData.
returnParentFieldValues	Version 2.1B only	returnParentFieldValues	New version implemented in lookupfield.p.
–	N/A	getComboBuffer	Returns a buffer to the ttDCombo TT.
–	N/A	getLookupBuffer	Same as above for ttLookup.
checkComboCacheData	No	checkComboCache	–
checkLookupCacheData	No	checkLookupCache	–
–	N/A	getComboBuffer	–
–	N/A	getLookupBuffer	–
retrieveSDFCache	No	retrieveData	Viewer handle only parameter
getAppserverData	No	retrieveAppserverData	–

**Table 1–1: Old field API to new field API map***(2 of 2)*

Old API	Continued support?	New API	Notes
lookupqp.p	No	fetchfield.p	New server procedure to support the new APIs.
stripLookupFields	No	—	Logic has been incorporated into checkLookupCache.

**Manuals:**     [\*Progress Dynamics Administration Guide\*](#)  
                       [\*Progress Dynamics ADM2 API Reference\*](#)

### 1.3 New application data caching on the client side

To further improve the performance of Progress Dynamics, Version 2.1B enhances the client-side data handling of Progress® SmartDataObjects™ (SDOs), dynamic combos, and dynamic lookups. New caching-related features include:

- SDOs now have built-in caching for application data on the client side.
- The existing client-side caching of dynamic lookups and combos has also been enhanced.
- Dynamic combos have the ability to take an SDO as a data source, and thereby benefit from the SDO-based caching, if it is enabled on that SDO.
- A new session parameter allows you to control whether you want to enable or disable caching for dynamic combos and lookups.

A new chapter in the [\*Progress Dynamics Programming Handbook\*](#) provides detail on all these features.

**Manuals:**     [\*Progress Dynamics Programming Handbook\*](#)

### 1.4 Manager API changes

For Progress Dynamics Version 2.1B, the PUBLIC API list for most Managers has changed. For more information, see the section on manager API changes in [Chapter 2, “Feature Comparisons.”](#)

<b>Manual:</b> <i><a href="#">Progress Dynamics Managers API Reference</a></i>
--

### 1.5 Repository schema changes

The Repository schema shown in the *[Progress Dynamics Repository Reference](#)* for Progress Dynamics Version 2.1B includes the schema updates through `icfdb020027delta.df`. For more information, see the section on Repository schema changes in [Chapter 2, “Feature Comparisons.”](#)

There has also been an ERwin version change. To work with the Repository’s ERwin model, you must upgrade to ERwin Version 4.1 SP2 (Build 4.1.2771).

<b>Manual:</b> <i><a href="#">Progress Dynamics Repository Reference</a></i>
--

### 1.6 ADM2 changes

For Progress Dynamics Version 2.1B, several API calls and properties have been added or changed. For more information, see the section on ADM2 changes in [Chapter 2, “Feature Comparisons.”](#)

<b>Manual:</b> <i><a href="#">Progress Dynamics ADM2 API Reference</a></i>
--

## 1.7 Web application development

Version 2.1B adds support for the following features in Web applications:

- TreeViews are now supported. You can use data nodes, menu nodes, and extract program nodes.
- Comments are now supported (but comments on comments are not supported).

**NOTE:** Mozilla 1.7 is now a supported browser for Progress Dynamics Web application users.

<b>Manual:</b> <i><a href="#">Progress Dynamics Web Development Guide</a></i>
---

## 1.8 Translation

Version 2.1B adds support for entity translation by providing DataField as a widget type option in the Translation Maintenance window. Previously, this window supported translations for a field label, depending on the widget type used to display the field. The DataField widget type option indicates that the translation for the specified field label is displayed for that field when referenced as an entity, such as when it is displayed in the Dynamics standard Find/Filter browser.

Support for entity translations also includes a Migrate Widget Translation tool that allows you to generate entity-based translations from existing widget-based translations that are already applied to fields in an application.

<b>Manuals:</b> <i><a href="#">Progress Dynamics Administration Guide</a></i>
---

## 1.9 Deployment

Version 2.1B adds these deployment features:

- A batch-mode Dynamics Configuration Utility (DCU) that allows you to create an unattended end-user Dynamics application installation or upgrade procedure.
- Site Data Dump and Load utilities that allow you to retain site-specific data during a Repository upgrade.
- Enhancements for deploying multi-transaction sequences.
- A significant run-time performance optimization for dynamic objects by allowing you to generate and deploy static-4GL objects that are functionally equivalent to the corresponding dynamic objects defined in the Repository, with minimal, if any, impact on development.
- Enhancements that facilitate the deployment of application objects between Progress Dynamics versions.

### 1.9.1 Batch-mode Dynamics Configuration Utility

The batch-mode DCU allows you to provide an unattended installation for your Progress Dynamics applications on end-user sites.

To use this new utility:

- 1 ♦ Set up and run the DCU GUI as usual, but parameterized to generate installation data instead of directly updating an installed Repository.
- 2 ♦ Write a batch command file or script that executes batch-mode DCU components, which use the generated data to update an installed Repository.



### **1.9.2 Site Data Dump and Load utilities**

When you upgrade an end-user site Repository you can use the new Site Data Dump and Load utilities to save out site-specific data, such as data for menus and other application features unique to that site, and reload the data after the Repository upgrade has completed. You can configure and run these utilities as part of the DCU or using a stand-alone GUI.

### **1.9.3 Multi-transaction sequence deployment**

Version 2.1B changes the default mechanism for deploying an application with multi-transaction sequences in order to prevent overwriting the current value of a sequence.

### **1.9.4 Static-4GL equivalents of dynamic objects**

Progress Dynamics provides a new 4GL Generator tool that allows you to generate and deploy static-4GL equivalents of dynamic objects. These static-4GL objects avoid some major performance bottlenecks of their dynamic equivalents by eliminating the need to:

- Download object definitions from the Repository server over the network.
- Dynamically construct an object from its definition for every instance of the same object rendered on the client.

Dynamics supports these generated static-4GL objects as true deployment versions of their dynamic equivalents that share the same registration in the Repository. Thus, you can easily maintain the same object as dynamic in the design-time Repository and deploy it as static for use with a run-time Repository. You can generate static-4GL equivalents of all dynamic objects except SmartBusinessObjects and Dynamics Web objects.

### 1.9.5 Deployment between versions

Progress Dynamics provides enhancements to the deployment tools and a recommended protocol for deploying application objects between development and run-time environments that are maintained separately between Version 2.1B and Release 10. To support inter-version deployment, the deployment tools support the following enhancements:

- The export and import functions of the deployment tools recognize deployment datasets (ADOs) from both release families.
- The dataset import functions ignore unknown tables (tables in the ADOs that do not exist in the target Repository).
- The dataset import functions ignore unknown fields (fields in import tables that do not exist in the same tables of the target Repository).
- The dataset import functions ignore unknown data types. That is, if a field in an import table has a data type not supported in the target Repository, the import functions ignore those fields.

<b>Manuals:</b> <i><a href="#">Progress Dynamics Administration Guide</a></i>
---

### 1.10 Print Preview

Version 2.1B enhances the Print Preview feature, allowing you to print to your choice of XML or HTML, in addition to Crystal Reports.

<b>Manuals:</b> <i><a href="#">Progress Dynamics Administration Guide</a></i>
---

## 1.11 Documentation set changes

Version 2.1B includes significant changes to documentation in addition to the changes required to document new and revised features, as shown in [Table 1–2](#).

**Table 1–2: Changes to documentation in Version 2.1B**

(1 of 2)

This manual...	Contains these changes...
<i>Progress Dynamics Administration Guide</i>	Adds Chapter 5, “Extending the Progress Dynamics Configuration Utility.” This chapter describes how to extend the Dynamics Configuration Utility (DCU) to install or upgrade your own applications at end-user sites. This includes information on how to create an unattended installation that runs the DCU in batch mode and how to save and restore site-specific Repository data that might otherwise be lost during an application upgrade.
	Adds Appendix D, “Deployment Notes.” This appendix provides information that supplements the deployment white paper on the Progress Software Development Network (PSDN) Web site ( <a href="http://psdn.progress.com">http://psdn.progress.com</a> ). This includes information on deploying: <ul style="list-style-type: none"> <li>• Multi-transaction sequences.</li> <li>• Static-4GL equivalents of dynamic objects.</li> <li>• Applications between different versions of Progress Dynamics.</li> </ul>
	Adds Appendix E, “Performance Notes.” This appendix includes the information provided in Section 3 of the Version 2.1A performance white paper on the PSDN Web site, updated for the current release.

**Table 1–2:      Changes to documentation in Version 2.1B** (2 of 2)

This manual...	Contains these changes...
<i>Progress Dynamics Programming Handbook</i>	Adds Chapter 2, “Extending Object Classes.” This chapter documents how to create custom objects by subclassing Dynamics classes.
	Adds Chapter 3, “Caching Application Data on the Client.” This chapter covers the new and enhanced application data-caching features in the framework.

---

## Feature Comparisons

This chapter provides comparisons about how the behavior of features in Progress Dynamics Version 2.1B differs from the previous release (Version 2.1A). It includes information on the Repository changes made to support the new features and enhancements. In addition, this chapter describes how changed features in Version 2.1B might affect the application code you wrote in Version 2.1A.

This chapter includes the following sections:

- [DCU on run-time client](#)
- [Clarification of Save Dynamic Object as Static menu command](#)
- [Extending object classes](#)
- [Manager API changes](#)
- [Repository schema changes](#)
- [ADM2 changes](#)

## 2.1 DCU on run-time client

In previous versions, the Progress Dynamics Configuration Utility (DCU) requires a compiler license to run. This requirement existed because some of the fix programs that the DCU applied needed to be compiled on the fly at run time.

Release 2.1B has removed this requirement. You can now run the DCU on a client site that does not have the Progress 4GL compiler installed.

## 2.2 Multi-transaction sequence deployment

In previous versions, Dynamics automatically deploys multi-transaction sequences as part of a general application deployment, but this automatic deployment can overwrite the current value of a sequence. To solve this problem, Version 2.1B changes some default settings for the sequence table, which prevents these sequences from being automatically deployed with an application. With this change, you must now deploy sequences explicitly in a separate deployment dataset.

## 2.3 Clarification of Save Dynamic Object as Static menu command

The Save Dynamic Object as Static menu command (on the File menu of the AppBuilder) was and still is intended to support the conversion of dynamic viewers and dynamic SDOs into static objects. It does not support other dynamic objects. For Version 2.1B, several bugs have been fixed to make the feature more robust. The *[Progress Dynamics Developer's Guide](#)* has been updated to better describe the feature and to provide guidance on handling the conversion of trigger and super procedure code. The *[Progress Dynamics Administration Guide](#)* contains new information on how to use this feature and compares it to the new 4GL Generator tool.

## 2.4 Extending object classes

In previous releases, to extend class behavior, you create a child class and insert it between existing Progress Dynamics classes. This is the only method of extending class behavior prior to Progress Dynamics Version 2.1B. With Version 2.1B, this method of changing class behavior is deprecated because such customizations are overwritten when you make a new deployment from a central repository.

Modifying class behavior by adding a custom class is the recommended approach. For more information on the new methods for extending classes, see the new chapter on extending object classes in the *[Progress Dynamics Programming Handbook](#)*.

## 2.5 Manager API changes

The Manager APIs change over time to support new functionality. To make it simpler for you to keep up with the changes, the following deprecation policy has been adopted.

### 2.5.1 Public and private APIs

The Progress Dynamics Managers contain many internal procedures and functions, but only a subset of them should be directly accessed by your applications.

To differentiate between APIs that should be accessed and those that should not be accessed, the header comment for each API call will include a line like one of the following:

```
ACCESS_LEVEL=PUBLIC
```

```
ACCESS_LEVEL=PRIVATE
```

API calls marked **PUBLIC** are subject to a formal deprecation policy. API calls marked **PRIVATE** are intended solely to support the framework's operation. They might be radically restructured or removed at any time to improve the framework's performance. There will be no formal notification of changes to **PRIVATE** API calls. API calls that have not been marked as either **PUBLIC** or **PRIVATE** are being evaluated. They will be marked in future releases.

### 2.5.2 Configuration File Manager

The following previously published API calls for this manager are now considered **PRIVATE**:

- buildErrorList
- setConfigManagerHandle
- setSystemParams
- setupPaths
- startEventHandler
- startManager
- startProcedure
- validateConfigFile

### **2.5.3 Connection and Service Type Managers**

The following sections list changes to the Connection and Service Type Managers' APIs.

#### **AppServer Connection Manager**

The following PUBLIC API calls have been added to the AppServer Connection Manager:

- findServiceRecord
- getConnectionParams
- getConnectionString
- getPhysicalService
- getServiceField
- getServiceHandle
- getServiceList
- isDefaultService
- setServiceHandle

#### **Database Connection Manager**

The following PUBLIC API calls have been added to the Database Connection Manager:

- findServiceRecord
- getConnectionParams
- getPhysicalService
- getServiceField
- getServiceHandle
- getServiceList
- isDefaultService
- registerService
- setServiceHandle



**JMS Connection Manager**

The JMS Connection Manager is no longer supported.

**2.5.4 Customization Manager**

The following previously published API calls for this manager are now considered PRIVATE:

- ICFCFM\_LoginComplete
- receiveCacheResultCodes
- receiveCacheTypeAPI

**2.5.5 General Manager**

The following previously published API calls for this manager are now considered PRIVATE:

- detectFileType
- getRecordUserPropx
- getSequenceValue
- getStatusRecord
- sendLoginCache

**2.5.6 Profile Manager**

The following PUBLIC API calls have been added to the Profile Manager:

- getProfileTTHandle
- recieveProfileCache

[Table 2–1](#) lists the changes to previously published API calls for this manager.

**Table 2–1: Changes to Profile Manager API calls**

API call	Change
sendLoginCache	Deprecated and removed from code.

## **2.5.7      Referential Integrity Manager**

The only PUBLIC API for this manager is the versionData procedure.

## **2.5.8      Repository Managers**

The following sections list changes to the Repository Managers' API calls.

### **Repository Manager**

The following PUBLIC API calls have been added to the Repository Manager:

- calculateObjectPaths
- destroyClassCache
- getClassFromInstance
- getClientCacheDir
- getMappedFilename

The following previously published API calls for this manager are now considered PRIVATE:

- buildAttributeList
- buildClassCache
- buildDenormalizedAttributes
- getAllObjectSuperProcedures
- getClassChildrenFromDB
- getClassChildrenProc
- getClassParentsFromDB
- getClassParentsProc
- getObjectPathedName
- getSDOincludeFile
- prepareInstance

Table 2–2 lists the changes to previously published API calls for this manager.

**Table 2–2: Changes to Repository Manager API calls**

API call	Change
areToolbarsCached	Deprecated and removed from code.
cacheObjectOnClient	Deprecated. Returns FALSE.
getCachePageInstanceBuffer	Deprecated. Returns UNKNOWN (?) value.
getCacheUiEventBuffer	Deprecated. Returns UNKNOWN (?) value.
getWhereConstantLevel	Deprecated. Returns a message.
getWhereStoredLevel	Deprecated. Returns a message.
IsA	Signature updated.
isObjectCached	Deprecated. Returns UNKNOWN (?) value.
launchClassObject	Deprecated. Returns UNKNOWN (?) value.
receiveCacheClass	Deprecated. Returns immediately.
receiveCacheMenu	Deprecated and removed from code.
receiveCacheObject	Deprecated.
retrieveClassCache	Deprecated. Returns an error.
serverFetchObject	Deprecated. Returns an error.
storeAttributeValues	It is now possible to add and update attribute values for attributes marked as runtime against a class.

### Repository Design Manager

The following PUBLIC API calls have been added to the Repository Design Manager:

- changeObjectType
- classHasAttribute
- copyObjectMaster

- generateClassCache
- getProductModuleList
- insertObjectPage
- prepareObjectName
- removeObjectPage
- removePageInstance
- setQualifiedTableName

The following previously published API calls for this manager are now considered PRIVATE:

- buildSchemaFieldTable
- cleanStoreAttributeValues
- editRyObjectInAB
- getComboBoxAttributes
- getCurrentProductModule
- getDupFieldCount
- getEditorAttributes
- getFuncLibHandle
- getIndexFields
- getIndexFieldsUnique
- getObjectTypeCodeFromDB
- getRadioSetAttributes
- getSelectionListAttributes
- getTableJoins
- openRyObjectAB
- removeInstances

- renameObjectInstance
- ripViewAsPhrase
- setCurrentProductModule

Table 2–3 lists the changes to previously published API calls for this manager.

**Table 2–3: Changes to Repository Design Manager API calls**

API call	Change
bufferRetrieveClassExtInfo	Deprecated. Returns a message.
cacheClassExtInfo	Deprecated. Returns a message.
cacheEntityDisplayField	Deprecated and removed from the code.
changeObjectInstance	Signature updated.
doServerRetrieveClassExtInfo	Deprecated. Returns a message.
generateDataFields	Signature updated.
getCacheClassExtBuffer	Deprecated. Returns a message.
insertObjectInstance	Signature updated.
insertObjectLinks	The dContainerObjObjectObj parameter is no longer used. But, it remains in the API signature for compatibility reasons.
productModuleList	Deprecated. Replaced by getProductModuleList.
registerSdoFields	Deprecated. Returns a message.
serverRetrieveClassExtInfo	Deprecated. Returns a message.

## 2.5.9 Security Manager

The following PUBLIC API calls have been added to the Security Manager:

- authenticateUser
- cacheGlobalSecurityAllocations

- `cacheGlobalSecurityStructures`
- `createGroupAllocation`
- `createGroupFromUser`
- `fieldandtokenSecurityCheck`
- `getContainerIcons`
- `menuItemSecurityCheck`
- `menuStructureSecurityCheck`
- `updateUserAllocations`

The following previously published API calls for this manager are now considered PRIVATE:

- `fieldSecurityCheck`
- `receiveCacheFldSecurity`
- `receiveCacheTokSecurity`

### **2.5.10 Session Manager**

The following previously published API calls for this manager are now considered PRIVATE:

- `buildPersistentProc`
- `clearActionUnderwayCache`
- `containerCacheUpFront`
- `deleteActiveSession`
- `deleteContext`
- `loginCacheAfter`
- `loginCacheUpfront`
- `loginGetClassCache`
- `loginGetMnemonicsCache`

- loginGetViewerCache
- parseAppServerInfo
- relogin

Table 2–4 lists the changes to previously published API calls for this manager.

**Table 2–4: Changes to Session Manager API calls**

API call	Change
containerCacheUpFront	Signature updated.
deleteContext	Signature updated.
increaseFrameWidth	Replaced with increaseFrameforPopup procedure.
loginCacheAfter	Signature updated.
resizeNormalFrame	Signature updated.
widgetWalk	Signature updated.  Also, in earlier versions, this procedure appends the pop-up handle to the widget's PRIVATE-DATA for static objects. The procedure no longer does this.

Table 2–5 changes to session properties.

**Table 2–5: Changes to Session Manager properties**

Session property	Change
ClassIgnoreContainedInstances	Deprecated.

## 2.5.11 User Interface Manager

The only PUBLIC API calls for this manager are the escapeData function and the setClientAction procedure.

### 2.5.12 Web Request Manager

The only PUBLIC API for this manager is the processRequest procedure.

## 2.6 Repository schema changes

You should be aware of the following changes in the schema of the Version 2.1B Repository.

### 2.6.1 New tables

The Version 2.1B Repository contains the following new tables:

- gsc\_data\_tag
- gsm\_tagged\_data

### 2.6.2 New fields

[Table 2–6](#) lists new fields in the Version 2.1B Repository.

**Table 2–6: New Repository fields**

Table	Field
gsc_object_type	custom_object_type_obj

### 2.6.3 Index changes

[Table 2–7](#) lists changes in the indexes of the Version 2.1B Repository.

**Table 2–7: Changes to Repository indexes**

Table	Index	Notes
gsc_object_type	XIE4gsc_object_type	New



## 2.7 ADM2 changes

You should be aware of the following changes to the Version 2.1B ADM2.

[Table 2–8](#) lists the new API calls.

**Table 2–8: New ADM2 API calls**

API call	Location
clearCache	cache.p
createDataSource	combo.p
displayField	combo.p lookup.p
findCacheItem	cache.p
notifyChildFields	lookupField.p
postCreateObjects	tvcontnr.p
prepareField	combo.p lookup.p
registerCacheItem	cache.p
resortQuery	data.p
retrieveBandsAndActions	toolbar.p
retrieveData	lookupField.p
returnParentFieldValues	lookupField.p

Table 2–9 lists the changed API calls.

**Table 2–9:      Changed ADM2 API calls**

API call	Location	Notes
canFindModRow	data.p	Deprecated.
constructObject	containr.p	Now checks for a generated object file first.
displayCombo	combo.p	Scheduled for deprecation.
displayCombo	lookup.p	Scheduled for deprecation.
refreshChildDependancies	combo.p	Scheduled for deprecation.

Table 2–10 lists the new ADM2 properties.

**Table 2–10:      New ADM2 properties**

Property	Class
CacheDuration	Data
ComboBuffer	Combo
DataSourceName	LookupField
KeepChildPositions	Viewer
LookupBuffer	Lookup
SchemaLocation	Data
ShareData	Data

Table 2–11 lists the changed ADM2 properties.

**Table 2–11:     Changed ADM2 properties**

Property	Class	Notes
ComboQuery	Combo	Scheduled for deprecation.
LookupQuery	Lookup	Scheduled for deprecation.

