

---

# Application Development Environment (ADE) Addenda

---

This web paper collects notes and enhancements about the application development environment for Progress® OpenEdge® Release 10. The paper covers the following topics:

- [ADM addenda](#)
- [Progress Dynamics addenda](#)
- [Progress WebSpeed addenda](#)
- [Third party acknowledgements](#)

---

© 2008 Progress Software Corporation. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

A (and design), Actional, Actional (and design), Allegrix, Allegrix (and design), Apama, Apama (and Design), Business Empowerment, DataDirect (and design), DataDirect Connect, DataDirect Connect64, DataDirect Technologies, DataDirect XQuery, DataXtend, Dynamic Routing Architecture, EasyAsk, EdgeXtend, Empowerment Center, Fathom, IntelliStream, Neon, Neon New Era of Networks, O (and design), ObjectStore, OpenEdge, PeerDirect, Persistence, POSSENET, Powered by Progress, PowerTier, Progress, Progress DataXtend, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Developers Network, Progress Sonic, ProVision, PS Select, SequeLink, Shadow, ShadowDirect, Shadow Interface, Shadow Web Interface, SOAPStation, Sonic, Sonic ESB, SonicMQ, Sonic Orchestration Server, Sonic Software (and design), SonicSynergy, SpeedScript, Stylus Studio, Technical Empowerment, WebSpeed, and Your Software, Our Technology—Experience the Connection are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, Apama Dashboard Studio, Apama Event Manager, Apama Event Modeler, Apama Event Store, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Cache-Forward, DataDirect Spy, DataDirect SupportLink, DataDirect XML Converters, Future Proof, Ghost Agents, GVAC, Looking Glass, ObjectCache, ObjectStore Inspector, ObjectStore Performance Expert, Pantero, POSSE, ProDataSet, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress RFID, PSE Pro, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Sonic Business Integration Suite, Sonic Process Manager, Sonic Collaboration Server, Sonic Continuous Availability Architecture, Sonic Database Service, Sonic Workbench, Sonic XML Server, The Brains Behind BAM, WebClient, Who Makes Progress, and Your World. Your SOA. 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 or trade names contained herein are the property of their respective owners.

**Third party acknowledgements** — See the “Third party acknowledgements” section on page 30.



November 2008

Last updated with new content: Release 10.2A

Product Code: 4496; R10.2A

For the latest documentation updates see the [OpenEdge Product Documentation](http://www.psdn.com/library/kbcategory.jspa?categoryID=129) category on PSDN (<http://www.psdn.com/library/kbcategory.jspa?categoryID=129>).

## ADM addenda

Since OpenEdge R10.1, the ADM documentation has the following addenda:

- Behavior change affecting procedure handle availability
- New DynCombo properties
- New DataView class
- Enabling columns for new records only
- Foreign values always applied when OpenOnInit is off
- Container publishes "createObjects" from createObjects
- Query manipulation on cached DataObjects
- Optimistic lock works for deletions
- NO-LOCK and read-only SDO tables
- EnableRule changes
- SDO always retrieves IndexInformation on first request

Also see the following addenda in the “Progress Dynamics addenda” section on page 10:

- Widget IDs for ADM2 and Progress Dynamics
- isObjQuoted function deprecated
- Changes to instanceOf function

### Behavior change affecting procedure handle availability

**Introduced:** OpenEdge R10.1A

Due to changes to support SmartDataFields, the AllFieldHandles and AllFieldNames properties no longer store procedure handles and names for contained objects on static containers. This information is available in the ContainerTarget and InstanceNames properties.

In previous versions, handle lists were built in the container's initializeObject procedure. The handle lists are now built during the container's createObjects procedure.

---

## New DynCombo properties

**Introduced:** OpenEdge R10.1A

The DynCombo class has two new properties, `AltValueOnAdd` and `AltValueOnRebuild`. These properties control the displayed value in the combo when the appropriate value cannot be determined.

The `AltValueOnAdd` property controls what to display if the data source's initial value for the object that is adding a record does not exist in the combo list. The `AltValueOnRebuild` property controls what to display if the combo list is rebuilt during editing, for example, when the parent field's value is changed.

Table 1 lists the allowed values for both properties.

**Table 1: DynCombo property values**

Value	Result
<Clear>	Clears the combo and displays nothing. This is the default value.
<First>	Displays the first value in the list.
<Last>	Displays the last value in the list.

There is no tool support for this properties. You must set them directly in the code.

## New DataView class

**Introduced:** OpenEdge R10.1A

A *data view* is a logical view of data. The structure of a data view does not reflect the structure of data in a particular data source. Rather, the data view's structure represents a convenient arrangement of data for a particular task. A data view might include calculated data or even data from several different data sources.

The OpenEdge `DataView` is an ADM2 implementation of a data view. The `DataView` is based on `ProDataSets` to take advantage of their efficiency. To make integrating `DataViews` into your applications easier, `DataViews` have the same object type as `SmartDataObjects` (SDOs). This type enables existing visual objects to connect to `DataViews` exactly as they do to SDOs.

When you look at how a data object gets data from the data source to the UI, the process has these main parts:

- The interface between the data object and the UI
- The in-memory data storage
- The data access from the data source

In the SDO, these parts are encapsulated in a single object. When you work with DataViews, the process is spread between several objects. The DataView handles the *data interface* with the UI. The data interface transfers data between the UI and the in-memory data storage. One or more ProDataSets, managed by a DataContainer, handle the in-memory data storage. Finally, a Service Interface that you create serves as the gateway to the back-end process that handles data access with the ultimate data source.

This separation fits well with the OpenEdge Reference Architecture style of programming. The reference architecture also calls for a Service Interface between the back-end processes of the Business Services layer and the front-end processes of the Presentation and Integration layers. The DataView fits neatly into this prescription.

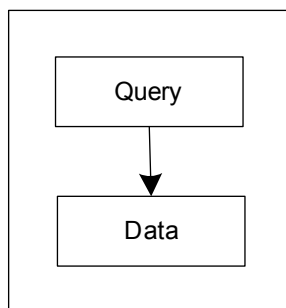
For more information on using DataViews, see the whitepapers in the *Implementing the OpenEdge Reference Architecture* section on PSDN:

<http://www.psdn.com/library/kbcategory.jspa?categoryID=289>.

The ADM2 API is considered self-documenting. Please see the source code in the *OpenEdge-install-dir\src\adm2* directory for details on new APIs and properties.

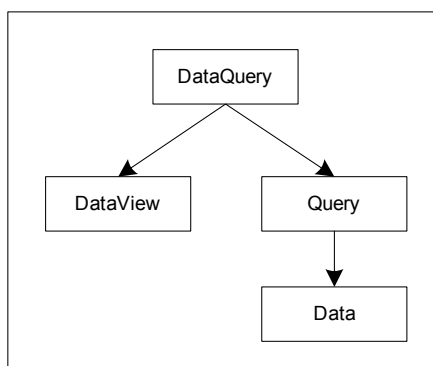
### Changes to the ADM2 hierarchy

Figure 1 shows the data object hierarchy in previous ADM versions.



**Figure 1:** Hierarchy before DataView

Figure 2 shows the new hierarchy.



**Figure 2:** Hierarchy with DataView

Changes in the query hierarchy have moved many existing APIs. The first instance of some APIs moved up the hierarchy to a different file. In some of these cases, an override was left in the original file; in other cases, the API was removed from that file. Consult the source code if you need to find an API that moved.

---

In addition, some classes were subdivided into multiple superprocedures:

- **Query** — Contained in `query.p` and `queryext.p`
- **Data** — Contained in `data.p`, `datacols.p`, `dataext.p`, and `dataextapi.p`

### Query property changes

Some existing properties have changed, as described in [Table 2](#).

**Table 2: New or changed properties**

API	Change
AutoCommit	This property now has Read/Write capability.
KeyFields	This property is Read only in the DataView class.
QuerySort	<p>When you specify sort direction in a BY phrase within this property, you can use a pseudo-keyword, <b>TOGGLE</b>. The <code>setQuerySort</code> function has internal logic to handle this word in a BY phrase. When the function finds <b>TOGGLE</b> in a BY phrase, the function checks the sort direction of the current BY phrase. The function applies the opposite sort direction to the new BY phrase.</p> <hr/> <p><b>Caution:</b> <b>TOGGLE</b> is not a true Progress 4GL keyword. You cannot use this word in BY phrases outside the context of this function and a few related functions.</p> <hr/>
UpdatableColumns	This property is Read/Write in the DataView class, but Read only in the Data class.

## Enabling columns for new records only

**Introduced:** OpenEdge R10.1C

SmartDataObjects and DataViews have a new `UpdatableWhenNew` property that accepts a comma-separated list of columns that can be updated only for new records.

DataVisual objects support a corresponding new `EnabledWhenNew` property that specifies visual fields that should be enabled only in add or copy mode. The `DataVisual` property inherits the value from the `UpdateTarget` by default, but can also be set to a comma-separated list of fields.

## Foreign values always applied when OpenOnInit is off

**Introduced:** OpenEdge R10.1C

Foreign values are now always applied on initialization. In previous versions, this did not happen when `OpenOnInit` was off. The `ForeignValues` would be added in `dataAvailable` when the parent navigated or as soon as any request was issued to the `AppServer`. But, a call to `openQuery` on a local connection could read all data. If you actually want to read all data in a child SDO, then the link to the data source should be removed.

## Container publishes "createObjects" from createObjects

**Introduced:** OpenEdge R10.1C

The ADM2 container now publishes "createObjects" from `createObjects`. Contained objects can subscribe to this event to ensure that their `createObjects` are done before `initializeObject` is called.

In Release 10.1C, this applies to toolbars and containers, but any ADM2 object can now implement a `createObjects` method and add the method to its `ContainerSourceEvents` in order to create widgets or read `Repository` data before `initializeObject` is called.

The container class has always subscribed to this event and child containers in `Progress Dynamics` have always relied on this behavior to ensure that all objects are created before resizing takes place. However, this is a behavior change for ADM2 non-window containers, like `SmartFrames` and `Viewers`, since their static container typically did not publish the "createObjects" event beforehand and their `createObjects` were called from `initializeObject`.

If you have customized code that relies on a late call to `createObjects`, there are two options to fix the problem:

- Move the customization to `initializeObject`. This is the recommended solution. The assumption is that, if a customization fails when `createObjects` is called before initialization, then the customization really belongs in `initializeObject`.
- Unsubscribe these containers from the "createObjects" event. You can do this by adding the following code in the main block of the object. You could also add this in `containrcustom.i` if it is general:

```
RUN modifyListProperty IN TARGET-PROCEDURE
    (TARGET-PROCEDURE,
     'Remove':U,
     'ContainerSourceEvents',
     'createObjects').
```

In environments with `Progress Dynamics`, this logic also needs checks to avoid that unsubscribing to the event which `Progress Dynamics` expects. For example, you could bypass the unsubscribe if `getUseRepository` is true.

---

## Query manipulation on cached DataObjects

**Introduced:** OpenEdge R10.1C

DataObjects that use caching now support client-side query manipulation. This enables Progress Dynamics filter support and enables you to use ADM2's filter source with cached SDOs.

Note that only criteria added with `assignQuerySelection` are applied to the cached SDO on the client. The `addQueryWhere` and `setQueryWhere` methods have no effect on the cached query.

## Optimistic lock works for deletions

**Introduced:** OpenEdge R10.1C

The optimistic conflict check is now done for deletions. Because of this, the record is refreshed from server and can be deleted on next attempt.

To accomplish the check, `fetchDBRowForUpdate` (which fetches the record and also performs the delete) now calls `compareDBRow`. In previous releases, `compareDbRow` was called after `fetchDBRowForUpdate` had completed. While these procedures generally are not intended for direct calls and local overrides, they can have general application-specific customizations which this change might affect. The most important change of behavior to consider is that `compareDBRow` now is called for deletes.

## NO-LOCK and read-only SDO tables

**Introduced:** OpenEdge R10.1C

You can use the `NoLockReadOnlyTables` property to specify a comma-separated list of read-only (non-updatable) tables. The specified tables remain NO-LOCKed during the transaction and are excluded from the optimistic lock check for changes of the SDO. Setting this property to 'ALL' specifies all the read-only tables in the SDO.

## EnableRule changes

**Introduced:** OpenEdge R10.1C

Several changes have been made to the toolbars' `EnableRule`.

### Find and filter actions `EnableRule` extended to disable when no parent

The "Find" and "Filter2" actions used on the various Browse\* toolbars, as well as the standard ADM2 "Filter" action, now check for "RecordState=RecordAvailable, NoRecordAvailable". The check ensures that the actions are disabled when no parent is available. The check identifies this by the `RecordState` in the target being "NoRecordAvailableExt".

For Progress Dynamics, "Find", "Filter", and "Filter2" are changed in the Repository. For standard ADM2, the `initAction` in `toolbar.p` is changed to add the rule to "Filter".

## New canUpdate() function for tableio and browse toolbar actions

All actions that can update data now include a `canUpdate()` function in their `EnableRule`. The `canUpdate()` function in the `datavisual` class (`adm2/datavis.p`) reflects whether the object can update data. The change ensures that the "Delete" action is disabled when there is no `UpdateTarget`. This change was added to all actions for consistency.

For standard ADM2, the `initAction` in `toolbar.p` now has `canUpdate()` in the `EnableRule` for "Add", "Copy", "Delete", and "Update".

The following table lists the actions changed in the Progress Dynamic's Repository:

Category	Actions
Tableio	"Add", "Copy", "Delete", "FolderUpdate", and "Update"
Other (for Browse* toolbars)	"Add2", "Copy2", "Delete2", and "Modify"

Actions like "Save", "Reset", and "Undo" do not have the new function. Those actions already have rules to ensure that they are never enabled until an appropriate action has been executed.

## SDO always retrieves IndexInformation on first request

**Introduced:** OpenEdge R10.1C

In previous releases, the SDO retrieves the `IndexInformation` property from the server only when the SDO is batching. Now, the SDO always retrieves the property from the server on the first request. Also note that the property only applies to a single table. In previous releases, the property contained information on all tables joined in the SDO. You can override `getIndexInfoTables` if you want it to return information for more tables.

Because of the changes to the `IndexInformation` property, an SDO on a client without batching now retrieves more context information on the first request. The context information for even a single table might be large and affect performance.

---

## Progress Dynamics addenda

Since OpenEdge R10.1, the Progress Dynamics® documentation has the following addenda:

- Generated toolbars and translations
- New log file for Deploy Static Objects tool
- Decimal values in dynamic combo-boxes
- New 4GL Generator options
- Objects without deployment types
- Context help for SmartDataFields
- General Manager API behavior change
- Interrupting triggers on dynamic viewers
- Translating Filter/Find windows
- Toolbar and Menu Designer enhancement
- Relogon and Session Reset features deprecated
- Maximum tables in dynamic lookup query
- Widget IDs for ADM2 and Progress Dynamics
- Simplified message dialog for Progress Dynamics
- isObjQuoted function deprecated
- Global translations for TAB and TITLE elements
- Customizing AppBuilder traffic to the Repository
- Changes to instanceOf function

### Generated toolbars and translations

**Introduced:** OpenEdge R10.1A

On the **Options** tab of the **4GL Generator** tool, you can select whether or not to include translations in your generated objects. If you choose not to include translations in a generated toolbar, the toolbar incurs an extra AppServer hit to retrieve translations from the Repository. Typically, the toolbar will produce two AppServer hits, one for translation and one for security. This behavior might degrade performance compared to dynamically creating toolbars. The size and complexity of the toolbar affect whether a dynamic or static toolbar will yield the better performance.

## New log file for Deploy Static Objects tool

**Introduced:** OpenEdge R10.1A

The **Deploy Static Objects** tool now enables you to produce a listing file. When the tool packages a deployment, the file lists all the packaged files.

By default, the tool creates a file named `listing.log` in your working directory. If a filename is not specified, no listing file is created.

The file contains the following fields, tab-delimited:

- File name
- Relative path
- Deployment type
- Design-only flag

## Decimal values in dynamic combo-boxes

**Introduced:** OpenEdge R10.1A

In previous releases, you might encounter problems developing decimal dynamic combo-boxes. Previous releases stored the default flag values in the format used during development. This prevented deployment to numeric formats other than the one used for development. When deployed with another numeric format, either the combo displayed no data and generated errors or incorrect data was stored in the Repository for the <None> or <A11> option.

Now default flag values are stored with American numeric format. You should enter the default flag values in the **SmartDataField Maintenance Tool** using the session's format, but the tool replaces the numeric decimal point with a period in the stored value. At run time, any stored periods in the data are replaced with the current session's numeric decimal point.

The DCU upgrade runs a fix program to convert existing default flag values for decimal dynamic combos to American format. The DCU must run in the format that was used to develop dynamic combos.

The fix program writes a message to the log file for each default flag value it attempts to convert. It writes a message when a value is converted successfully or writes a message if the conversion fails. Review the DCU log file after the upgrade and manually correct any failures. Any failures that are not corrected might not behave properly at run time.

---

**Note:** There are several Progress Dynamics objects in the Repository that store blank default flag value instance attributes and give messages in the log file. These are for the `cbSCMTool` dynamic combo on the following viewer objects: `gsmsxgenvi ewv`, `gsmsxotvi ewv`, `gsmsxpmvi ewv`. The messages for these objects are expected and you can ignore them.

---

You must manually convert any dynamic combo instance attributes stored in static viewer code in your application to American numeric format.

---

## New 4GL Generator options

**Introduced:** OpenEdge R10.1A

The **4GL Generator** tool enables you to create static equivalents of your dynamic objects to optimize performance. The **4GL Generator** now enables you to set the GENERATE-MD5 and MIN-SIZE options for the r-code it generates.

## Objects without deployment types

**Introduced:** OpenEdge R10.1A

The **Deploy Static Objects** tool now enables you to specify what to do with objects that do not have a deployment type. The **Include objects with blank deployment types** toggle box controls whether or not these objects are included in your deployment. By default, this option is set and any object with a blank deployment type is included in the deployment. When the toggle is unchecked, these objects are excluded from the deployment.

## Context help for SmartDataFields

**Introduced:** OpenEdge R10.1A

When mapping context help in previous versions for SmartDataFields, there was a mismatch between the data required to create the mapping and the data required to retrieve the context help. In OpenEdge R10.1A, you map the help in the **Map Context Help** tool with the same data that you use to retrieve the help. [Table 3](#) shows the new values for mapping SDF context help. These values are consistent with the values used to map context help for viewer datafields.

**Table 3: Mapping SDF context help**

Field	Value
Help container filename	Container name
Help object filename	Viewer name
Help fieldname	SDF FieldName

## General Manager API behavior change

**Introduced:** OpenEdge R10.1A

The `updateTableViaSDO` procedure in the General Manager has changed. In previous releases, this procedure ran SDOs directly. The procedure now starts SDOs using the `startDataObject` procedure. The procedure now can support updates with both static and dynamics SDOs. However, unlike in previous versions, the procedure cannot start an SDO that has not been registered in the Repository.

## Interrupting triggers on dynamic viewers

**Introduced:** OpenEdge R10.1A

You should use RETURN ERROR instead of RETURN NO-APPLY to interrupt triggers on dynamic viewers. RETURN NO-APPLY does not propagate up the procedure stack. This behavior means that it cannot interrupt a default event in a procedure that is defined as a trigger event for a widget on a dynamic viewer. An example of this situation is when you want to avoid setting focus on the next field on the TAB event.

In addition, RETURN ERROR works if the event procedure is defined directly as a persistent trigger outside of Progress Dynamics.

## Translating Filter/Find windows

**Introduced:** OpenEdge R10.1A

Providing translations for Progress Dynamics **Filter/Find** windows require several steps using the **Translate** window, the **Menu Item Translation** window, and the **Message Control** window.

A **Filter/Find** window contains the following translatable components:

- The window title
- Menu items and their tooltips
- Tab folder labels
- Radio-set labels
- Browse column labels

---

**Note:** The browse picks up any translations for database column labels that are added through the **Entity Translation** window.

---

- Labels of the widgets on the **Advanced** tab for filter windows
- The error message indicating when a permanent filter is set

---

## Where to translate items

Table 4 lists the tools for translating items and which components you translate through those tools.

**Table 4: Tools for translating window components**

Tool	Component
Translate window	The window title
	Tab folder labels
	Radio-set labels
	Browse column labels
	Labels of widgets on the <b>Advanced</b> tab for filter windows
Menu Item Translation window	Menu items and their tooltips
Message Control window	Warning AF:149, the message when a permanent filter is set

### Translating the window title

Providing a translation of the **TITLE** widget's **Original Label** does not yield a properly translated window title. The window title consists of the actual name of the tool or table and one of two hidden fields, `fiFilterLabel` and `fiFindLabel`. The hidden fields provide the appropriate word, "Filter" or "Find", depending on the mode in which the **Filter/Find** window is launched. That word is then added to the **TITLE** widget's label to produce the final title.

To properly translate the window title, you need to provide the following translations:

- A translation for `fiFilterLabel`
- A translation for `fiFindLabel`
- A translation for the **TITLE** widget

---

**Note:** You should only translate the words after the hyphen. The rest of the label is added as appropriate by the framework.

---

## Toolbar and Menu Designer enhancement

**Introduced:** OpenEdge R10.1A

You can now specify an internal procedure that is published just before the creation of a menu item or toolbar button. The **Create event** field in the **Other** section of an **Item** property sheet controls this behavior. The toolbar or menu looks for the procedure in the object specified in the **Item link** field. If no **Item link** is specified, the toolbar or menu looks in the container's super procedure.

If you want to create a user-defined list of menu items, you can define the menu items using the `defineAction` function. For example, if you have a sub-menu called 'Favorites', you can call the `defineAction` function in the procedure specified in the **Create event** field to add child items, as shown in the following code:

```
DYNAMIC-FUNCTION('defineAction' in SOURCE-PROCEDURE,
  'Sol',
  'Name,Caption,OnChoose,Parent,Type',
  'Sol'+ CHR(1)+'Solitaire'+CHR(1)+'runSolitaire'+ CHR(1)+'Favorites'
  +CHR(1) + 'RUN').
```

If you want to create a user-defined list of menu items that are re-constructed each time the menu is selected (for example, a list of opened windows), you can define a 'Menu drop function' for an item. This is a function located in the linked object defined by the 'Item link' (or container's super if the item link is blank) that returns a `CHR(1)` delimited list of menu item names and captions.

Suppose you have a sub-menu called 'Modules' that changes with the context of the current record. You can add the following code to the function:

```
cTarget = DYNAMIC-FUNCTION('linkHandles':U IN TARGET-PROCEDURE,
  'ContainerToolbar-Source':U) NO-ERROR.
hTarget = WIDGET-HANDLE(cTarget).
DYNAMIC-FUNCTION("defineAction" IN hToolbar,
  "Favorites",
  "OnChoose",
  "RunFavorites").
cList = "OE,Order Entry,IN,Inventory,CM,Customer Maintenance"
      = REPLACE(cList," ",CHR(1)).
RETURN cList.
```

The procedure 'RunFavorites' defined on the parent item 'Favorites' would require a character input parameter. The system would pass the Key (odd entry in the list) to the procedure. (For more information, see the *Working with the version 9 ADM: Using and Customizing the SmartToolbar* whitepaper.)

## Relogon and Session Reset features deprecated

**Introduced:** OpenEdge R10.1C

The **Relogon** and **Session Reset** menu options are deprecated. They have been removed from the menus. However, the underlying APIs are still in the code for any user who has customized or extended these features.

## Maximum tables in dynamic lookup query

**Introduced:** OpenEdge R10.1C

In the *OpenEdge Development:Progress Dynamics Basic Development* manual, the section on defining dynamic lookups states that a lookup's query can contain a maximum of 10 tables. The lookup's query can now contain a maximum of 18 tables.

---

## Widget IDs for ADM2 and Progress Dynamics

**Introduced:** OpenEdge R10.1C

When the OpenEdge GUI client renders widgets on a window, it calls a Microsoft Windows API during the widget creation. The Windows function allows the caller to pass a child-window identifier which Windows uses to identify that given widget. By default, the OpenEdge GUI client generates a new identifier for each widget each time the window runs. While this process is efficient, it prevents the use of third-party automated testing tools that rely on your widgets having the same unique identifier across sessions.

OpenEdge R10.1A introduced the `WIDGET-ID` attribute to enable you to assign a fixed identifier to the widgets on your interface. This attribute can provide third-party tools with the unique identifiers they require for automated testing. That release also included the **Widget ID Assignment Utility** PRO\*Tool to enable you to assign widget IDs to simple GUI containers without opening them. However, that tool writes the widget ID directly into the static code for each widget which prevents it from assigning widget IDs to certain widgets.

---

**Note:** To make use of the `WIDGET-ID` attribute, you must start your session with the Use Widget ID (`-usewidgetid`) startup parameter. Using widget IDs causes extra processing to render your widgets. Since widget IDs are only intended for use with automated testing tools, you can avoid any performance impact by not specifying this parameter for your deployed applications.

---

OpenEdge R10.1C extends the support for widget IDs. New AppBuilder preferences enable automatic assignment of the `WIDGET-ID` attribute for certain widgets. This release also introduces the **Runtime WIDGET-ID Assignment Tool** PRO\*Tool to enable you to easily assign widget IDs in ADM2 and Progress Dynamics applications. You can now assign widget IDs to the following:

- Widgets in dynamic objects, for example, a dynamic SmartDataViewer
- Dynamically-rendered widgets in SmartObjects, for example, a SmartSelect
- Multiple instances of a SmartObject in the same run-time container

---

**Note:** Some widgets are composed of elements that the third-party testing tools consider separate widgets. The widget IDs for these elements are always assigned by the AVM when the widget is rendered. For example, the AVM assigns widget IDs to the columns of a browse by incrementing the browse's widget ID by a constant for each column.

---

The **Runtime WIDGET-ID Assignment Tool** creates an XML file to store the widget IDs that it assigns. When the AVM renders the widgets at run time, it uses the numbers from this file to assign values to the widget's `WIDGET-ID` attributes. The XML file creates a context in which a dynamically-generated widget can have the same widget ID over several sessions.

As long as you do not change the number of widgets on a container, you can edit the container and get a repeatable set of unique widget IDs. However, if you add or remove a widget on a container, you must generate a new XML file. A given widget's widget ID might not match between the old and new files. Regenerating the file creates a new context, and you need to record your tests again in the third party tool to make use of these new widget IDs.

If you start a session without the Use Widget ID (`-usewidgetid`) startup parameter, the ADM rendering engine ignores the `WIDGET-ID` attribute and assigns identifiers through its default process. If you do specify the startup parameter, the ADM uses the following precedence to assign widget IDs:

1. The ADM reads the `widgetIDFileName` property for the container. If the ADM finds a filename, it uses the values in the XML file to assign the `WIDGET-ID` attribute for the listed widgets.
2. For the remaining widgets, the ADM determines if a `WIDGET-ID` attribute was specified in a static file. If the ADM finds a hard-coded value, it assigns that value to the widget.
3. For any remaining widgets, it assigns the next unused value, working from the top-end of the range backwards. The `WIDGET-ID` attribute accepts even `INTEGER` values of 2 through 65534.

---

**Note:** Widget IDs must be unique across a window. When a conflict occurs, a warning dialog appears and the ADM assigns the next available value as described in [Step 3](#).

---

As this process shows, the actual value of a widget ID might be assigned by one of several methods. In general, the user does not need to know the actual value of a widget ID. The third-party tools access these numbers through the Windows API, rather than directly from OpenEdge. A user never needs to prepare a complete list of all the widget IDs in an application. The only time a user cares about the value is if a conflict arises because a newly assigned widget ID is not unique. The user can then alter the value of that widget to create a unique widget ID.

### AppBuilder preferences

The Appbuilder **Preferences** dialog includes a **Widget ID** tab that enables you to control how the widget ID features operate:

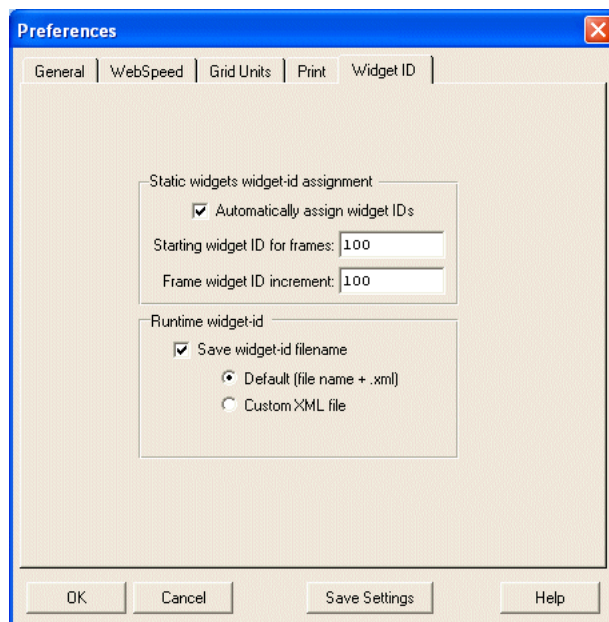


Table 5 describes this preference tab.

**Table 5: Widget ID Preferences**

<b>Control</b>	<b>Description</b>
<b>Automatically assign widget IDs</b> toggle	When selected (the default), the AppBuilder automatically assigns WIDGET-ID attribute values to any static widget when it is created.
<b>Starting widget ID for frames</b> fill-in	Defines the value for the first in a container when the AppBuilder assigns the WIDGET-ID attribute automatically.
<b>Frame widget ID increment</b> fill-in	Defines the amount by which the AppBuilder increments the value for each additional frame in a container.
<b>Save widget-id filename</b> toggle	When selected (the default), the AppBuilder automatically assigns a value for the <code>widgetIDfileName</code> property when you first save a container. This setting has no effect after the first time you save the container. The AppBuilder stores the value in the static code or in the Progress Dynamics Repository, as appropriate.
<b>Default</b> radio button	Defines the <code>widgetIDfileName</code> property as the default value, <code>container_filename.xml</code> . For example, the default XML file for the window, <code>wCust.w</code> , is <code>wCust.xml</code> .
<b>Custom XML file</b> radio button	Displays a fill-in field where you can specify an alternate name for the XML file.

The preference options in the top box only apply when the AppBuilder automatically assigns widget IDs to the widgets in a static container. The preference options in the bottom box only apply to the XML files used by the **Runtime WIDGET-ID Assignment Tool**.

### Widget ID numbering

The tools increment the widget IDs differently, depending on which tool you use. When assigning static widget IDs, the AppBuilder or **Widget ID Assignment Utility** use the following simple pattern:

1. The widget ID of the first frame in the container is the value specified in the **Starting widget ID for frames** preference.
2. The widget ID for the first widget in that frame is 2.
3. The widget ID for each additional widget in that frame increments by 2.
4. When all widgets in the frame have a widget ID, the tool moves to the next frame.
5. The tool sets the widget ID of the next frame by incrementing the widget ID of the previous frame by the value in the **Frame widget ID increment** preference.

6. The tool repeats [Step 2](#) through [Step 4](#) until all the widgets have widget IDs.
7. When the ADM renders the object, it adds the widget IDs of the widget and the frame together to produce the final widget ID value. For example, using the default values, the first widget in the first frame has a widget ID of 102; the second widget is 104; the first widget in the second frame is 202; the second widget is 204; and so on.

The **Runtime WIDGET-ID Assignment Tool** uses a different scheme when assigning widget IDs to widgets. Each object type has a different increment assigned in the `adecomm/_widgaps.p` procedure. As the tool assigns widget IDs in a container, it adds the widget ID of the preceding widget and the increment for the preceding widget's object type together to calculate the widget ID of the current object. Some widgets also act as containers for other widgets. For these widgets, the widget ID of the first contained widget is always zero plus the widget's increment. When the ADM renders the container at runtime, it assigns the final widget ID for the contained widgets by adding the container's widget ID to the widget ID of the contained widget. [Table 6](#) lists the default increments used by the tool.

**Table 6: Runtime WIDGET-ID Assignment Tool default increments**

Object type	Increment
SmartContainer static FRAME	200
SmartBrowser or DynBrow	250
SmartViewer or DynView	1000
DataField (DynView only)	4
SmartDataField	20
DynCombo/DynLookup	10
SmartFilter	100
SmartFilter Field	6
SmartFrame	2000
SmartToolbar	200
SmartToolbarActions	2
SmartSelect	10
SmartFolder	1000
SmartFolder Page	20
SmartLOBField	10
SmartPanel	50
TreeNode	2000

---

**Note:** If these increments are not wide enough to create unique widget IDs for your application's widgets, you can customize the `adecomm/_widgaps.p` procedure as needed. However, remember that the `WIDGET-ID` attribute only accepts even `INTEGER` values of 2 through 65534.

---

For example, for a static `SmartWindow` with a `SmartToolbar`, `SmartDataBrowser`, `SmartDataViewer`, and a `SmartFilter`, the tool generates the following widget IDs:

1. The widget ID of the `SmartWindow`'s default frame is **200**, by default.
2. The tool adds the frame's widget ID, **200**, to the frame's increment, **200**, to create the next widget ID, **400**. It assigns **400** as the widget ID of the first widget on the default frame, the `SmartToolbar`.
3. The tool adds the `SmartToolbar`'s widget ID, **400**, to its increment, **200**, to create the next widget ID, **600**. The `SmartToolbar` is a container for its `SmartToolbarActions`, which visualize as buttons. So, the buttons are on the `SmartToolbar`'s frame, rather than the `SmartWindow`'s default frame. The tool skips the buttons and assigns **600** to the next widget on the default frame, the `SDB`.
4. The `SDB`'s widget ID and its increment, **250**, makes the next widget ID, **850**. The tool assigns **850** to the next widget on the default frame, the `SDV`.
5. The `SDV`'s widget ID and its increment, **1000**, makes the next widget ID, **1850**. The tool assigns **1850** to the last widget on the default frame, the `SmartFilter`.
6. The tool moves to the next frame on the `SmartWindow`, the `SmartToolbar`'s frame. The tool starts at **0** and adds **2**, the increment for the first contained widget. Thus the first button's widget ID is **2** in the XML file. When the ADM renders the object, it adds the `SmartToolbar`'s widget ID to the button's widget ID to get a final widget ID of **402**.

---

**Note:** Menu options cannot be assigned widget IDs; only the buttons can have widget IDs.

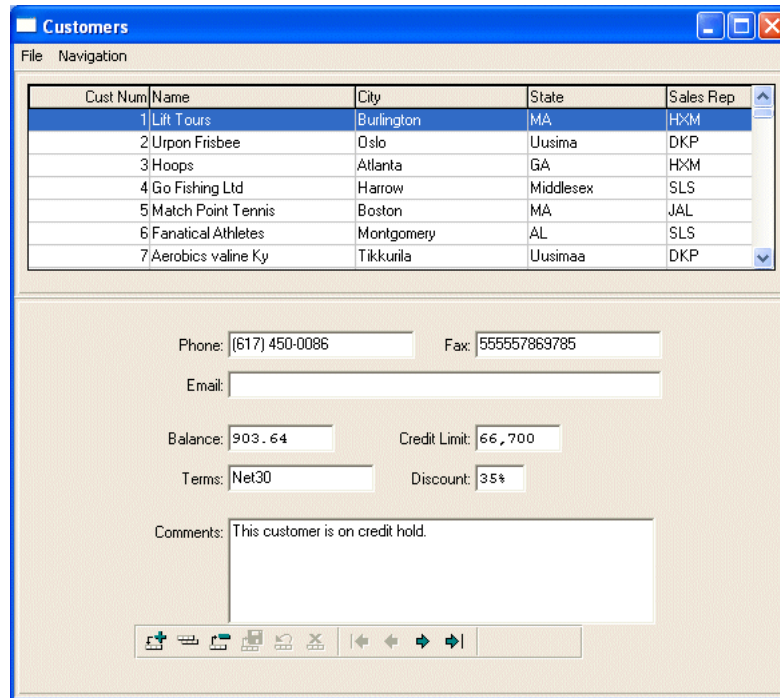
---


7. The tool assigns widget IDs to all the buttons on the `SmartToolbar` and then moves on through the remaining frames. In this example, it assigns widget IDs for the columns in the `SDB`, the widgets on the `SDV`, and the fields on the `SmartFilter`.

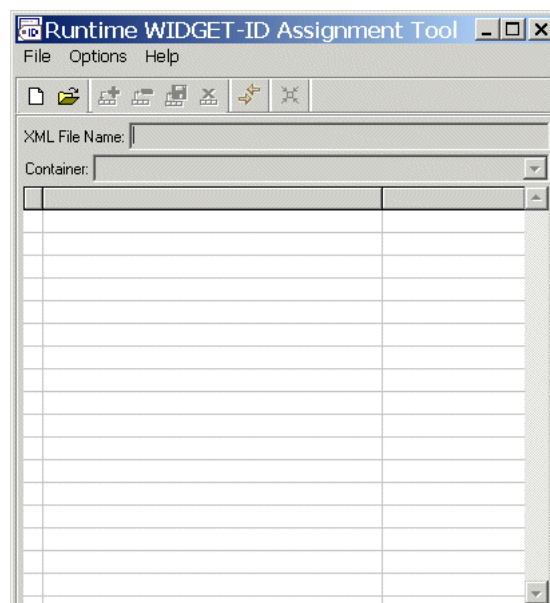
## Using the Runtime WIDGET-ID Assignment Tool

You create widget ID files for your containers with the **Runtime WIDGET-ID Assignment Tool**. The general procedure for creating a widget ID file is as follows:

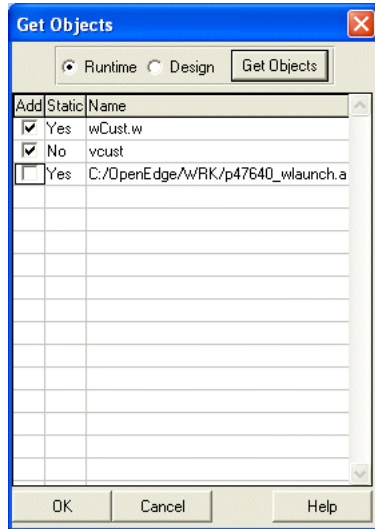
1. Launch the container from another running container in the AppBuilder. This workaround bypasses a limitation caused by how the AppBuilder runs windows. For example, take this **Customers** window:



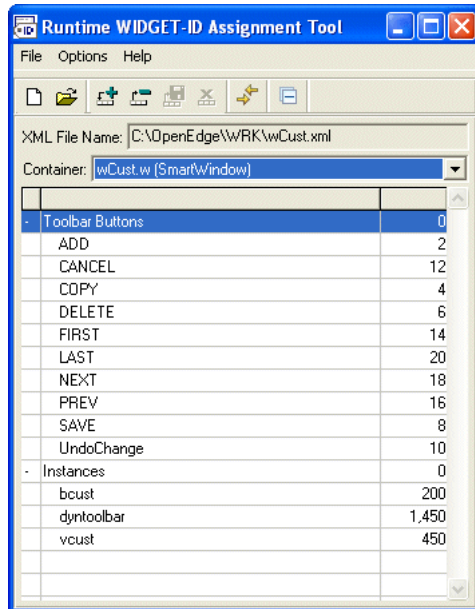
2. Click the **Runtime WIDGET-ID Assignment**  button on the PRO\*Tool palette. The **Runtime WIDGET-ID Assignment Tool** window appears:



3. Create a new XML file.
4. Click **Add Containers to XML file**. The Get Objects dialog appears.
5. Choose **Runtime** and click **Get Objects**. The **Get Objects** dialog box displays all the running containers:



6. Deselect the temporary AppBuilder version of the launch window and click **OK**. The tool assigns widget IDs to any widgets that do not have a widget ID and saves them to the XML file. It displays the widget IDs in a hierarchical browse:



7. If you change the container later, you can reopen the file and click **Synchronize XML data with source object** to update the widget IDs. The browse marks any objects that you added or removed from the container.
8. If you get a conflict warning for one of the widget IDs, you can enter a new value in the browse and save the new value back to the XML file.

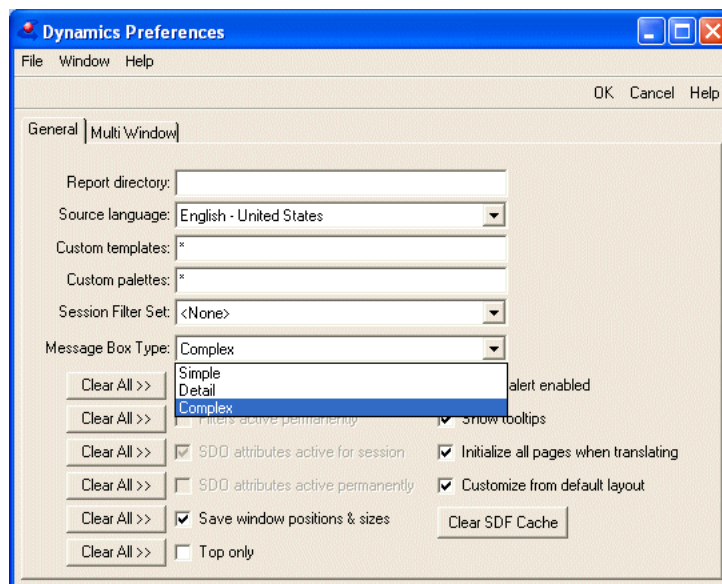
## Simplified message dialog for Progress Dynamics

**Introduced:** OpenEdge R10.1C

OpenEdge R10.1C adds a method to control the visualization of the standard Progress Dynamics message dialog. A new Progress Dynamics session parameter, `MessageBoxType`, enables you to choose between three levels of visualization. So, without changing any objects in your application, you can change the functionality of the standard message dialog between different session types by changing the value of the session parameter.

By default, Progress Dynamics sessions continue to use the full functionality of the standard message dialog. None of the shipped session types include the `MessageBoxType` session parameter in their definitions. If you want to use simplified message dialogs, you should add the session parameter in your custom session types.

Users can override the session parameter in the user preferences. On the Administration window, choose **File**→**Preferences** to launch the **Dynamics Preferences** dialog. The **Message Box Type** combo-box enables users to choose the style of message box for their session.



The standard message dialog appears when the application code calls either the Session Manager's `showMessages()` or `askQuestion()` internal procedures. The Session Manager's `showWarningMessages()` internal procedure launches a different dialog box. The new session parameter does not affect the visualization of this alternate message box.

---

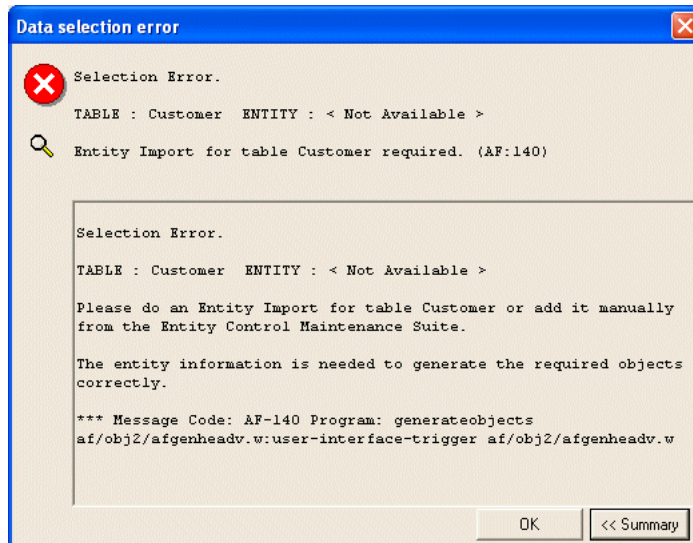
## Visualization levels

The MessageBoxType session parameter accepts the following values:

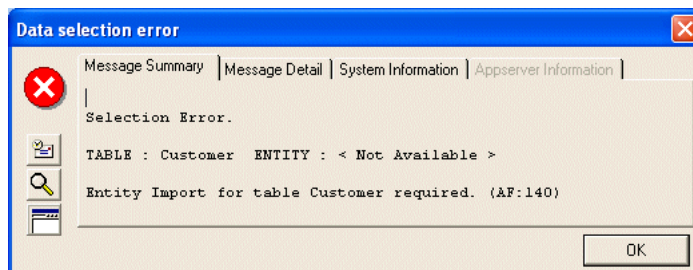
- **Simple** — Shows only the **Message Summary** of the standard message dialog:



- **Detail** — Shows the **Message Summary** and adds a **Details** button. Clicking **Details**, expands the message box to show an editor containing the **Message Detail** information of the standard message dialog:



- **Complex** — Shows the standard message box:



## Interaction with SESSION:DEBUG-ALERT

When SESSION:DEBUG-ALERT is TRUE, the standard message dialog includes a button to access an AVM stack trace through a secondary ABL message box. When you use either the Simple or Detail visualization level, the message dialog displays a similar button. While the icon is different, the functionality is the same.

## isObjQuoted function deprecated

**Introduced:** OpenEdge R10.1C

The `isObjQuoted` functions in the `ADM2 smart.p` class and the Progress Dynamics Session Manager have been deprecated. These functions were used internally by `fixQueryString`, which was deprecated in OpenEdge R10.1B.

## Global translations for TAB and TITLE elements

**Introduced:** OpenEdge R10.1C

Static containers can use global translations for TAB and TITLE elements, as well as named (container-specific) translations. The global translations are used on every window (for title) or every tab folder that has not been translated explicitly.

## Customizing AppBuilder traffic to the Repository

**Introduced:** OpenEdge R10.1C

Progress Dynamics code should not have include references to `adeuib`, although calls to `adeuib` procedures can be made. However, in previous releases, some `adeuib` includes were erroneously added to the Progress Dynamics source code. OpenEdge R10.1C removes these include references.

The AppBuilder uses a procedure, `ry/prc/rydynsckrp.p`, to read the information that it needs to edit dynamic objects. It writes data back into the Repository using `ry/prc/rygendynp.p`. In previous releases, these procedures had references to include files in the `adeuib` directory. Because the `adeuib` source code is no longer shipped with the product, these references made it impossible to compile the procedures as released. The `adeuib` source code is still available from PSDN.

Release 10.1C moves the working code for reading and writing dynamic objects to `adeuib/_dynsckr.p` and `adeuib/_gendyn.p`, respectively. These procedures are called from `rydynsckrp.p` and `rygendynp.p`. All of the calls that the AppBuilder makes to this functionality are still through `rydynsckrp.p` and `rygendynp.p` (and not the `adeuib` procedures). This allows customers who have customized `rydynsckrp.p` or `rygendynp.p` to keep their customizations.

The only way to extend `rydynsckrp.p` or `rygendynp.p` is to modify the code itself; there is no defined way to extend these procedures. Note that customizing these procedures is not recommended. Customers who have existing customizations should overwrite `rydynsckrp.p` or `rygendynp.p` with their customized code. This code should be compared with the PSDN versions of `_dynsckr.p` or `_gendyn.p` to determine what has changed.

## Changes to instanceOf function

**Introduced:** OpenEdge R10.1C

The `instanceOf` function has been extended to also work for non-Progress Dynamics applications. This enables class type checks to be done transparently across the two framework variants in the cases where the `ObjectType` does not provide sufficient separation. The `ObjectType` is still supported and used throughout the ADM2 and the AppBuilder.

---

The `instanceOf` in ADM2 uses the super procedure hierarchy. It considers the class name to be the super procedure name without the path and the `.p` extension. It is actually derived from the `&ADMsuper` preprocessor, which matches the super procedure name in all shipped super procedures. Progress Dynamics uses the actual class names in the Repository.

The `instanceOf` function also allows transparent references to classes that have different names in the two models, but really are the same class. For example, both `instanceOf('containr')` and `instanceOf('container')` can be used transparently in both environments. The following table lists names that can be used interchangeably:

<b>ADM2</b>	<b>Progress Dynamics</b>
Smart	Base (not likely to be used as all objects are of this type)
Combo	DynCombo
Lookup	DynLookup
Datavis	DataVisual
Containr	Container

Progress Dynamics has more classes than ADM2. References to classes that do not exist in ADM2, like "dynSDO", return false in ADM2.

The `instanceOf` function still checks the `ObjectType` as a last resort. This is not new behavior, but something that one should be aware of as there are classes in Progress Dynamics that have similar names to some of the `ObjectTypes`. When `InstanceOf` checks with these names, it likely gives the desired effect because Progress Dynamics names are based on the ADM2, but you should use some caution. In particular, avoid using `instanceOf` to check for classes that begins with "Smart", as these names typically are classes in Progress Dynamics and `ObjectTypes` in ADM2.

## Progress WebSpeed addenda

Since OpenEdge R10.1, the WebSpeed® documentation has the following addenda:

- [WebSpeed support for BLOBs](#)

### WebSpeed support for BLOBs

**Introduced:** OpenEdge R10.1A

Previous OpenEdge releases introduced the BLOB data type and enabled WebSpeed applications to include BLOBs in their output. OpenEdge now supports the upload of BLOBs to WebSpeed applications.

To protect your system from malicious code, you should not allow unrestricted processing of uploaded binary files. Therefore, WebSpeed does not automatically store any file uploaded as part of a Web request. Uploaded files are held in memory. Your application can then manipulate the file through a MEMPTR.

The following code is an example of how to create a form that can upload a file:

```
<HTML>
  <BODY>
    <FORM ENCTYPE="multipart/form-data"
      ACTION="http://yourhost/msngr_path/msngr/target_app" METHOD="POST">
      <INPUT type="file" name="attachment1"> <INPUT type="submit">
    </FORM>
  </BODY>
</HTML>
```

To enable a WebSpeed application to accept data from this form, you must do several things:

- Upgrade your WebSpeed Messenger to OpenEdge R10.1A. The Messengers in previous versions cannot accept BLOBs.
- Specify a maximum size for uploaded files.
- Write code to manipulate the uploaded file through its MEMPTR. This code should ensure that the uploaded file contains no malicious content before it is stored to disk. WebSpeed provides a new API function to return the file's MEMPTR. See the [“get-binary-data”](#) section on page 28 for the API's description.

#### Specifying maximum size of uploaded files

Uploading a file as part of a Web request ties up the WebSpeed agent that processes the request. This might cause performance problems for your application if several large files are uploaded at the same time. The BLOB data type has a maximum size of 1 GB, so a single upload could lock an agent for a significant period.

To control this problem, you can specify a maximum size for uploaded BLOBs. The maximum size is set with the `binaryUploadMaxSize` property in the `ubroker.properties` file. You can set it on the **Agent>Advanced Features** page of the **WebSpeed Transaction Server's Properties** sheet, as shown:

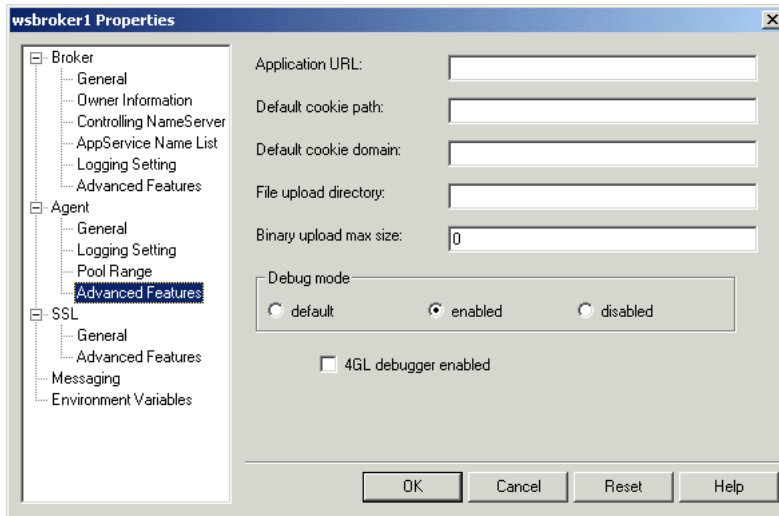


Table 7 lists the results of various settings for the upload maximum size.

**Table 7: binaryUploadMaxSize settings**

Value	Result
Any negative value	The agent does not limit the size of uploaded BLOBs.
0	The agent does not accept uploaded BLOBs.
1 byte – 1GB	The agent accepts uploaded BLOBs up to this size.

**Note:** The limit applies separately to each file received in a single Web request. If your Web page can accept more than one file in a particular Web request, you should consider the performance impact of several files of that size being uploaded together.

### get-binary-data

This function returns a MEMPTR to the contents of a file posted from a Web page by a multipart/form-data form. The function uses the GET-BINARY-DATA method of the WEB-CONTEXT system handle.

**Location:** `web\method\cgiutils.i`

#### Parameters:

INPUT `p_name` AS CHARACTER

The name of a field on the form. The field's type must be `file`.

**Returns:** MEMPTR

**Note:** If the uploaded file exceeds the limit set in the `binaryUploadMaxSize` property, the function returns the Unknown value (?) instead of a `MEMPTR`.

**Example:**

```
/* This example shows how to retrieve the data from a file passed on a given
request, and save it as a file with the same name to the WebSpeed's working
directory. Note that 'filename' is the name of the field in the form posted. */

DEFINE VAR mFile AS MEMPTR NO-UNDO.
DEFINE VAR cfile AS CHAR NO-UNDO.

/* 'filename' refers to the name of the field in the form. get-binary-data
returns the contents of the file associated with the form field named
'filename'. */
ASSIGN mFile = get-binary-data("filename").

IF mFile <> ? THEN DO:
  /* if we got a valid pointer, save data to file. The value of the field
'filename' is the file name posted */
  ASSIGN cfile = get-value("filename").

  COPY-LOB FROM mFile TO FILE cfile NO-CONVERT.

END.
```

---

## Third party acknowledgements

OpenEdge includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999 The Apache Software Foundation. All rights reserved (Xerces C++ Parser (XML) and Xerces2 Java Parser (XML)); Copyright © 1999-2002 The Apache Software Foundation. All rights reserved (Xerces Parser (XML)); and Copyright © 2000-2003 The Apache Software Foundation. All rights reserved (Ant). The names “Apache,” “Xerces,” “ANT,” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “Apache” , nor may “Apache” appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact [apache@apache.org](mailto:apache@apache.org). Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes software copyrighted by DataDirect Technologies Corp., 1991-2007.

OpenEdge includes DataDirect products for the Microsoft SQL Server database which contains a licensed implementation of the Microsoft TDS Protocol.

OpenEdge includes software authored by David M. Gay. Copyright © 1991, 2000, 2001 by Lucent Technologies (dtoa.c); Copyright © 1991, 1996 by Lucent Technologies (g\_fmt.c); and Copyright © 1991 by Lucent Technologies (rnd\_prod.s). Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

OpenEdge includes software authored by David M. Gay. Copyright © 1998-2001 by Lucent Technologies All Rights Reserved (decstrtod.c; strtodg.c); Copyright © 1998, 2000 by Lucent Technologies All Rights Reserved (decstrtof.c; strtord.c); Copyright © 1998 by Lucent Technologies All Rights Reserved (dmisc.c; gdtoa.h; gethex.c; gmisc.c; sum.c); Copyright © 1998, 1999 by Lucent Technologies All Rights Reserved (gdtoa.c; misc.c; smisc.c; ulp.c); Copyright © 1998-2000 by Lucent Technologies All Rights Reserved (gdtoaimp.h); Copyright © 2000 by Lucent Technologies All Rights Reserved (hd\_init.c). Full copies of these licenses can be found in the installation directory, in the c:/OpenEdge/licenses folder. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS

ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

OpenEdge includes software developed by IBM. Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder. OpenEdge includes ICU software 1.8 and later - Copyright © 1995-2003 International Business Machines Corporation and others All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

OpenEdge includes Infragistics NetAdvantage for .NET v2008 Vol 1 CLR 2.0. Copyright © 1996-2008 Infragistics, Inc. All rights reserved.

OpenEdge includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). Copyright © 1998-2007 The OpenSSL Project. All rights reserved. This product includes cryptographic software written by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)). This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)). Copyright © 1995-1998 Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)) All rights reserved. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org). Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at <http://oss.software.ibm.com/icu4j/>.

OpenEdge includes the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright ©1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

OpenEdge includes Imaging Technology copyrighted by Snowbound Software 1993-2003. [www.snowbound.com](http://www.snowbound.com).

OpenEdge includes Sonic software, which includes software developed by Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999-2000 The Apache Software Foundation. All rights reserved. The names "Ant", "Axis", "Xalan", "FOP", "The Jakarta

---

Project”, “Tomcat”, “Xerces” and/or “Apache Software Foundation” must not be used to endorse or promote products derived from the Product without prior written permission. Any product derived from the Product may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission. For written permission, please contact [apache@apache.org](mailto:apache@apache.org).

OpenEdge includes Sonic software, which includes software Copyright © 1999 CERN - European Organization for Nuclear Research. Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

OpenEdge includes Sonic software, which includes software developed by ExoLab Project (<http://www.exolab.org/>). Copyright © 2000 Intalio Inc. All rights reserved. The names “Castor” and/or “ExoLab” must not be used to endorse or promote products derived from the Products without prior written permission. For written permission, please contact [info@exolab.org](mailto:info@exolab.org). Exolab, Castor and Intalio are trademarks of Intalio Inc.

OpenEdge includes Sonic software, which includes the JMX Technology from Sun Microsystems, Inc. Use and Distribution is subject to the Sun Community Source License available at <http://sun.com/software/communitysource>.

OpenEdge includes Sonic software, which includes software developed by the ModelObjects Group (<http://www.modelobjects.com>). Copyright © 2000-2001 ModelObjects Group. All rights reserved. The name “ModelObjects” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “ModelObjects”, nor may “ModelObjects” appear in their name, without prior written permission. For written permission, please contact [djacobs@modelobjects.com](mailto:djacobs@modelobjects.com).

OpenEdge includes Sonic software, which includes code licensed from Mort Bay Consulting Pty. Ltd. The Jetty Package is Copyright © 1998 Mort Bay Consulting Pty. Ltd. (Australia) and others.

OpenEdge includes Sonic software, which includes files that are subject to the Netscape Public License Version 1.1 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>. Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is Mozilla Communicator client code, released March 31, 1998. The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright © 1998-1999 Netscape Communications Corporation. All Rights Reserved.

OpenEdge includes Sonic software, which includes software developed by the University Corporation for Advanced Internet Development <http://www.ucaid.edu> Internet2 Project. Copyright © 2002 University Corporation for Advanced Internet Development, Inc. All rights reserved. Neither the name of OpenSAML nor the names of its contributors, nor Internet2, nor the University Corporation for Advanced Internet Development, Inc., nor UCAID may be used to endorse or promote products derived from this software and products derived from this software may not be called OpenSAML, Internet2, UCAID, or the University Corporation for Advanced Internet Development, nor may OpenSAML appear in their name without prior

written permission of the University Corporation for Advanced Internet Development. For written permission, please contact [opensaml@opensaml.org](mailto:opensaml@opensaml.org).

OpenEdge includes software Copyright © 2003-2006, Terence Parr All rights reserved. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes the UnixWare platform of Perl Runtime authored by Kiem-Phong Vo and David Korn. Copyright © 1991, 1996 by AT&T Labs. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T LABS MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

OpenEdge includes software developed by Vermont Creative Software. Copyright © 1988-1991 by Vermont Creative Software.

OpenEdge includes software developed by the World Wide Web Consortium. Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All rights reserved. This work is distributed under the W3C® Software License [<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>] in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

OpenEdge includes XML Tools, which includes versions 8.9 of the Saxon XSLT and XQuery Processor from Saxonica Limited (<http://www.saxonica.com/>) which are available from SourceForge (<http://sourceforge.net/projects/saxon/>). The Original Code of Saxon comprises all those components which are not explicitly attributed to other parties. The Initial Developer of the Original Code is Michael Kay. Until February 2001 Michael Kay was an employee of International Computers Limited (now part of Fujitsu Limited), and original code developed during that time was released under this license by permission from International Computers Limited. From February 2001 until February 2004 Michael Kay was an employee of Software AG, and code developed during that time was released under this license by permission from Software AG, acting as a "Contributor". Subsequent code has been developed by Saxonica Limited, of which Michael Kay is a Director, again acting as a "Contributor". A small number of modules, or enhancements to modules, have been developed by other individuals (either written especially for Saxon, or incorporated into Saxon having initially been released as part of another open source product). Such contributions are acknowledged individually in comments attached to the relevant code modules. All Rights Reserved. The contents of the Saxon files are subject to the Mozilla Public License Version 1.0 (the "License"); you may not use these files except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/> and a copy of the license can also be found in the installation directory, in the c:/OpenEdge/licenses folder. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

---

OpenEdge includes XML Tools, which includes Xs3P v1.1.3. The contents of this file are subject to the DSTC Public License (DPL) Version 1.1 (the "License"); you may not use this file except in compliance with the License. A copy of the license can be found in the installation directory, in the c:/OpenEdge/licenses folder. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is xs3p. The Initial Developer of the Original Code is DSTC. Portions created by DSTC are Copyright © 2001, 2002 DSTC Pty Ltd. All rights reserved.