



# ASNA MONARCH

## ASNA Monarch 2.0 Reference Manual

*A Reference Manual containing the ASNA  
Monarch Framework Classes and  
Members.*

*ASNA Monarch™ transforms iSeries applications  
originally written in ILE RPG or RPG/400, into a  
native Microsoft .NET application.*



### ASNA

**ASNA Inc. - USA**  
9901 West IH-10, Suite 1000  
San Antonio, Texas 78230  
USA

**800-289-ASNA (2762)**  
**210-408-0212**  
**210-408-0211 Fax**

**ASNA Ltd. - Europe**  
31 Frederick Sanger Road  
Surrey Research Park  
Guildford, Surrey GU2 7EP  
England

**+44 1483 570666**  
**+44 1483 578191 Fax**

*Written by: ASNA  
Documentation  
Team*

*August, 2005*

This page intentionally left blank.

# Table Of Contents

<b>ASNA Monarch Reference.....</b>	<b>1</b>
<b>Chapter 1 - !Pages .....</b>	<b>1</b>
!Diagnose.....	1
!Eoj .....	1
!ExpiredSession .....	2
<b>Chapter 2 - Class Library.....</b>	<b>3</b>
ASNA.Monarch.WebDspf.....	3
<b>ConditionalProperty Class .....</b>	<b>3</b>
ConditionalProperty Constructor().....	3
ConditionalProperty Constructor(string).....	3
Clone Method.....	4
GetValidValues Method .....	4
NewInstance Method .....	5
Reset Method.....	5
ToString Method .....	5
Property Property.....	5
PropString Property .....	6
<b>ConditionalValue Class .....</b>	<b>7</b>
ConditionalValue Constructor() .....	7
ConditionalValue Constructor(string, string).....	7
ToString Method .....	8
Condition Property .....	8
Value Property .....	9
<b>DdsCharField Class .....</b>	<b>11</b>
DdsCharField Constructor() .....	11
CheckedValue Property .....	11
DefaultValue Property.....	12
InputStyles Property.....	12
Length Property .....	13
Lower Property.....	13
UncheckedValue Property .....	13
Value Property .....	14
DdsConstant Class .....	15
DdsConstant Constructor() .....	15
EncodedLiteral Method.....	15
EncodedNonBlank Method .....	16
ResetText Method.....	16
Text Property .....	16
<b>DdsDecField Class.....</b>	<b>19</b>
DdsDecField Constructor().....	19
Decimals Property.....	19
EditCode Property.....	<b>Error! Bookmark not defined.</b>
EditWord Property.....	21
Length Property .....	25
Type Property .....	26

Value Property .....	26
<b>DdsField Class .....</b>	<b>27</b>
DdsField Constructor() .....	27
<b>Methods:</b>	
OnPreRender Method .....	27
ResetCheck Method .....	28
ResetColor Method .....	28
ResetErrorMessage Method .....	28
ResetErrorMessageId Method .....	29
ResetVisibleCondition Method .....	29
<b>Properties:</b>	
Check Property .....	29
Color Property .....	30
Comment Property .....	30
ErrorMessage Property .....	30
ErrorMessageId Property .....	31
VisibleCondition Property .....	31
IsDesignTime Field .....	31
<b>DdsFile Class.....</b>	<b>33</b>
DdsFile Constructor() .....	33
AddErrorMessage Method .....	34
LoadPostData Method .....	34
RaisePostDataChangedEvent Method .....	35
SetResponseIndicators Method .....	35
AttnID Property .....	35
AttnKeys Property .....	36
BackColor Property .....	36
BannerStyle Property .....	37
CursorField Property .....	38
CursorLocation Property .....	39
CursorRecord Property .....	39
ErrorSubfile Property .....	39
FuncKeys Property .....	40
PopUpWindows Property .....	40
ResultingIndicators Property .....	41
InputDataAvailable Field .....	41
<b>DdsInputField Class.....</b>	<b>43</b>
DdsInputField Constructor() .....	43
LoadPostData Method .....	43
RaisePostDataChangedEvent Method .....	44
RenderBeginTag Method .....	44
RenderEndTag Method .....	45
ResetMessageId Method .....	45
Alias Property .....	45
ChangeInd Property .....	46
MessageId Property .....	46
PositionCursor Property .....	46
Protect Property .....	47

---

Text Property .....	47
Usage Property .....	48
Values Property .....	48
ValuesStyle Property .....	49
ValuesText Property .....	50
VirtualRowCol Property.....	50
<b>DdsLink Class .....</b>	<b>53</b>
DdsLink Constructor() .....	53
ResetVisibleCondition Method .....	53
TextFieldLength Property.....	54
TextFieldName Property .....	54
TextValue Property .....	54
UrlFieldName Property .....	55
UrlFieldName Property .....	55
UrlValue Property.....	55
VirtualRowCol Property.....	56
VisibleCondition Property.....	56
<b>DdsRecord Class .....</b>	<b>57</b>
DdsRecord Constructor() .....	57
CollectResponseIndicators Method .....	57
GetConditionedValue Method .....	58
GetConditionedValue Method .....	58
GetDdsFile Method .....	59
GetResultIndicator Method .....	59
IsAidKeyInEffect Method.....	60
IsAidKeyInEffect Method.....	60
IsNotFalse Method(string).....	61
IsNotFalse Method(string, string) .....	61
IsTrue Method(string).....	62
IsTrue Method(string, string).....	62
OnPreRender Method.....	62
SetDataChanged Method.....	63
SetOptionIndicators Method.....	63
AttnKeys Property .....	64
ChangeInd Property.....	64
EraseFormats Property.....	65
FuncKeys Property .....	66
ReturnData Property .....	66
Window Property.....	67
WindowHeight Property .....	67
WindowLeft Property.....	67
WindowLeftField Property .....	68
WindowTitle Property .....	68
WindowTitleField Property .....	68
WindowTop Property .....	69
WindowTopField Property.....	69
WindowWidth Property .....	69

<b>DdsSubfile Class</b> .....	<b>71</b>
DdsSubfile Constructor().....	71
CreateChildControls Method.....	71
GetConditionedValue Method.....	72
IsNotFalse Method.....	72
IsTrue Method.....	73
LoadPostData Method.....	73
OnInit Method.....	74
RaisePostDataChangedEvent Method.....	74
ReadBrowser Method.....	74
RenderChildren Method.....	75
ResetNextChanged Method.....	75
NextChanged Property.....	75
recordCount Property.....	76
subfileStyle Property.....	76
<b>DdsSubfileControl Class</b> .....	<b>79</b>
DdsSubfileControl Constructor().....	79
OnInit Method.....	79
Render Method.....	80
ResetSubfileMessage Method.....	80
ResetSubfileMessageId Method.....	81
scrollDownControls Method.....	81
ClearRecords Property.....	81
ControlledRecord Property.....	82
DisplayFields Property.....	82
DisplayRecords Property.....	83
InitializeRecords Property.....	83
ProgramQLen Property.....	84
ProgramQ Property.....	84
SubfileMessageId Property.....	85
SubfileMessage Property.....	86
SubfilePage Property.....	87
SubfileSize Property.....	87
SubfileStyle Property.....	88
<b>EraseProperty Class</b> .....	<b>89</b>
EraseProperty Constructor().....	89
EraseProperty Constructor(String).....	89
GetValidValues Method.....	90
NewInstance Method.....	90
<b>ErrMsgIdProperty Class</b> .....	<b>91</b>
ErrMsgIdProperty Constructor().....	91
ErrMsgIdProperty Constructor(String).....	91
GetValidValues Method.....	92
NewInstance Method.....	92
<b>ErrMsgProperty Class</b> .....	<b>93</b>
ErrMsgProperty Constructor().....	93
ErrMsgProperty Constructor(String).....	93
GetValidValues Method.....	94

NewInstance Method .....	94
<b>KeyProperty Class .....</b>	<b>95</b>
KeyProperty Constructor() .....	95
KeyProperty Constructor(String) .....	95
KeyProperty Constructor(String) .....	96
GetValidValues Method .....	96
NewInstance Method .....	97
<b>Page Class .....</b>	<b>99</b>
Page Constructor () .....	99
GetFileName Method .....	99
OnCopyBrowserToDspFile Method .....	100
OnCopyDspFileToBrowser Method .....	101
OnInit Method .....	101
SaveViewState Method .....	102
DataSet Property .....	102
OutOfSequence Property .....	103
RunProgram Property .....	103
Device Field .....	104
DspF Field .....	104
<b>Enumerations .....</b>	<b>105</b>
BannerStyles Enumeration .....	105
DateTimeFormat Enumeration .....	106
DecType Enumeration .....	107
FieldUsages Enumeration .....	108
InputStyles Enumeration .....	109
SubfileStyles Enumeration .....	110
ValuesStyle Enumeration .....	111
ASNA.VisualRPG.Runtime Assembly .....	113
<b>CLProgram Class .....</b>	<b>113</b>
CLProgram Constructor() .....	114
AddLiblEntry Method(string) .....	114
AddLiblEntry Method(string, ASNA.DataGate.Client.LiblPosition, string) .....	115
Asterisk_BCat Method .....	116
Asterisk_TCat Method .....	116
ChangeDataArea Method(string, integer, integer, string) .....	117
ChangeDataArea Method(string, string) .....	117
ClearPFM Method .....	118
DeleteOverride Method(string) .....	119
DeleteOverride Method(string, ASNA.Monarch.OverrideScope) .....	119
InitializePFM Method(string, string, ASNA.Monarch.InitializePFMOptions, integer) .....	120
InitializePFM Method(string, string, integer) .....	121
OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object) .....	122
OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object, ASNA.Monarch.OverrideScope) .....	123
Percent_SST Method .....	123
Percent_Switch Method .....	124
RetrieveDataArea Method(string) .....	125
RetrieveDataArea Method (string,integer,integer) .....	125

RmvLiblEntry Method.....	126
RTVJOBA_DATE Method.....	126
RTVJOBA_JOB Method.....	126
RTVJOBA_NBR Method.....	127
RTVJOBA_SWS Method.....	127
RTVJOBA_TYPE Method.....	128
RTVJOBA_USER Method.....	128
<b>Job Class.....</b>	<b>129</b>
Job Constructor().....	130
ApplyOverrides Method (string, ASNA.VisualRPG.Runtime.DBfile).....	130
ApplyOverrides Method (string, ASNA.VisualRPG.Runtime.PrintFile).....	131
ApplyOverrides Method (string, ASNA.VisualRPG.Runtime.WorkStnFile).....	131
DeleteOverride Method(string).....	131
DeleteOverride Method(string, ASNA.Monarch.OverrideScope).....	132
DeleteOverride Method(string, integer).....	132
EndPrograms Method.....	133
FindProgram Method.....	133
getDatabase Method.....	133
GetInvokedMessageQueue(int).....	134
GetInvokedMessageQueue(string).....	134
GetLdaField Method.....	135
GetSwitches Method.....	135
GetSwitch Method.....	136
OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object).....	136
OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object, ASNA.Monarch.OverrideScope).....	137
PeekInvocation Method(int).....	137
PeekInvocation Method(String).....	138
PopInvocation Method.....	138
PushInvocation Method.....	138
RegisterProgram Method.....	139
SetLdaField Method.....	139
SetSwitch Method.....	140
ShutDown Method.....	140
UnregisterProgram Method.....	140
CurrentJob Property.....	141
Database Property.....	141
MessageFileFolder Property.....	141
StartupMoment Property.....	142
ExternalQueue Field.....	142
lda Field.....	142
messageFileFolder Field.....	143
PdsJobName Field.....	143
PdsJobNumber Field.....	143
PdsJobUser Field.....	144
<b>Program Class.....</b>	<b>145</b>
Program Constructor.....	146
DataArea_In Method.....	146

DataArea_Out Method .....	147
DataArea_Unlock Method .....	147
Dispose Method .....	148
ExecCommand Method .....	148
GetLdaField Method .....	148
RemoveMessage Method(string) .....	149
RemoveMessage Method(string, string) .....	149
SendExternalMessage Method .....	150
SendProgramMessage Method(string) .....	150
SendProgramMessage Method(string, string, ASNA.Monarch.MessageTypes)	151
SendProgramMessage Method(string, string, string) .....	152
SendProgramMessage Method(string, string, string, string) .....	152
SendProgramMessage Method(string, string, string, string, ASNA.Monarch.MessageTypes) .....	153
SetLdaField Method .....	154
UPDATE Property .....	154
UDAY Property .....	154
UMONTH Property .....	155
UYEAR Property .....	155
monarchJob Field .....	155
<b>WebDevice Class .....</b>	<b>157</b>
WebDevice (ASNA.Monarch.WebJob) .....	157
Attach Method .....	158
Attach Method .....	158
Detach Method(string) .....	159
Detach Method .....	159
Dispose Method .....	159
GetActiveFile Method .....	160
GetThreadDevice Method .....	160
Read Method(string) .....	160
Read Method(ASNA.Monarch.WebDisplayFile) .....	161
SignalDataReadyForProgram Method .....	161
SignalDataReadyForUser Method .....	162
WaitForDataForProgram Method .....	162
WaitForDataForUser Method .....	162
AbEnd Field .....	163
AbEndMessage Field .....	163
AbEndStack Field .....	164
DevicelsOpen Field .....	164
<b>WebDisplayFile Class .....</b>	<b>165</b>
WebDisplayFile Method .....	165
Close Method .....	166
GetCurrentRow Method .....	166
InitMessageSubfile Method(string, string) .....	166
InitMessageSubfile Method(string, string, char[ ]) .....	167
Open Method .....	167
Read Method .....	168
Update Method .....	168

---

Write Method.....	168
AttnID Property .....	169
FeedbackAID Property.....	169
FeedbackCursor Property .....	169
FeedbackField Property.....	170
FeedbackFlags Property.....	170
ResultingIndicators Property .....	170
Device Field .....	171
<b>WebJob Class.....</b>	<b>173</b>
WebJob Method () .....	173
EndPrograms Method .....	174
ExecuteStartupProgram Method.....	174
PopInvocation Method .....	174
PushInvocation Method .....	175
ShutDown Method .....	175
Start Method .....	175
CurrentWebJob Property .....	176
Device Field .....	176
<b>Enumerations .....</b>	<b>177</b>
MessageTypes Enumeration .....	177
OverrideOptions Enumeration.....	178
OverrideScope Enumeration.....	180
<b>Chapter 3 - Web Server Controls.....</b>	<b>181</b>
DdsCharField Class .....	181
DdsDecField Class .....	181
DdsFile Class.....	181
DdsInputField Class.....	182
DdsLink Class.....	182
DdsRecord Class.....	182
DdsSubfile Class.....	183
DdsSubfileControl Class .....	183
<b>Chapter 4 - Operation Codes.....</b>	<b>185</b>
<b>New Op Codes.....</b>	<b>185</b>
DCLWORKSTNFILE .....	185
EXFMT.....	186
RETURN .....	187
<b>Improved Op Codes .....</b>	<b>189</b>
CALL - Call a Program.....	189
PARM - Identify Parameters .....	191

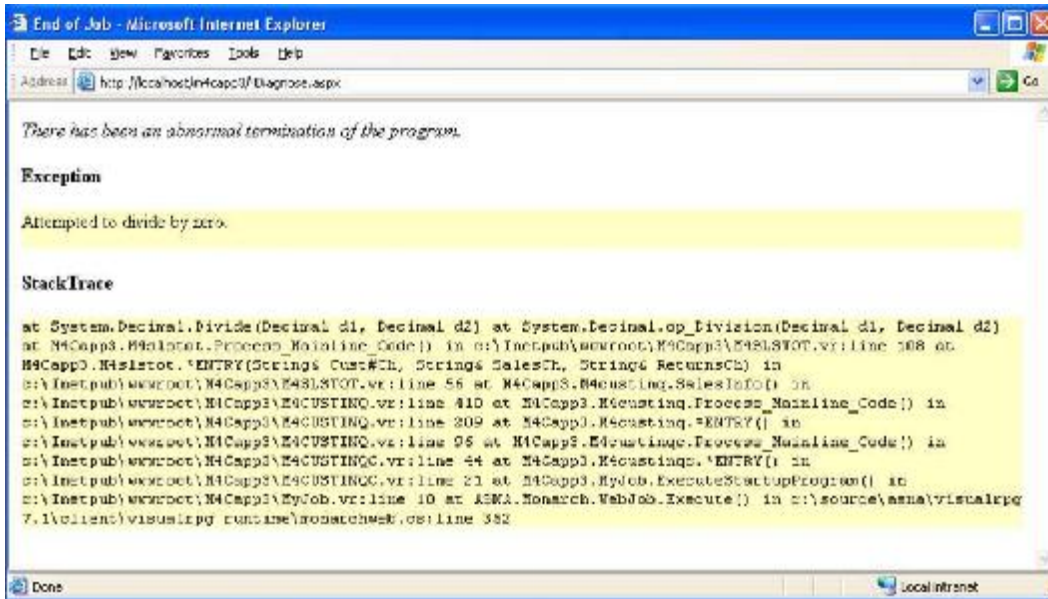
# ASNA Monarch Reference

## Chapter 1 - !Pages

### !Diagnose

The **!Diagnose** page is an end-point for crashed sessions. It displays the crashing exception and call stack.

Page\_Load should abandon the session. **!Diagnose** is delivered as **!Diagnose.Aspx.Vr** and **!Diagnose.Aspx** (example shown below).



### See Also

!EoJ | !ExpiredSession

### !EoJ

The **!EoJ** is an application **End of Job** page. Use **!EoJ** to redirect the user to the main menu or home page, as shown in the picture below. **!EoJ** is replaceable by the Migrator and is delivered as **!EoJ.Aspx.Vr** and **!EoJ.Aspx**.



### Example

```
BegSr Page_Load Event(*this.Load)
DclSrParm Name(Sender) Type (*Object)
DclSrParm Name(3) Type (System.EventArgs)
    Session.Abandon()
EndSr
```

### See Also

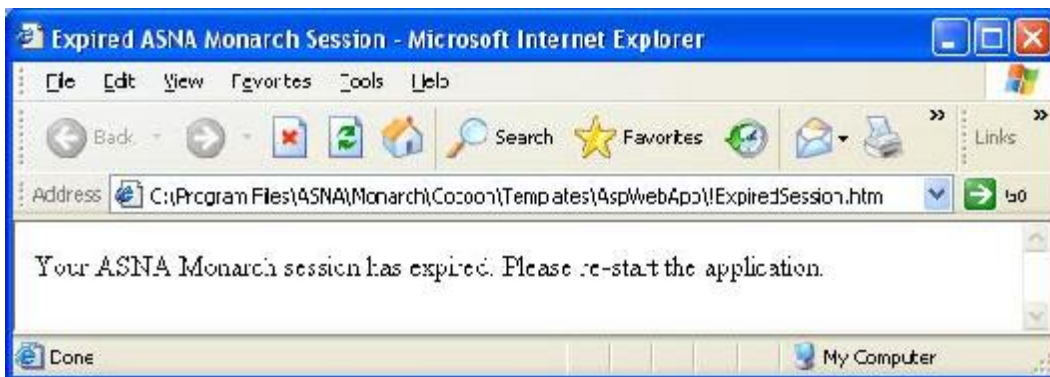
!Diagnose | !ExpiredSession

---

## !ExpiredSession

The **!ExpiredSession** application end-point for expired sessions. **!ExpiredSession** is not an ASPX page to avoid starting a new session.

!ExpiredSession is delivered as !ExpiredSession.htm.



### See Also

!Diagnose | !Eoj

---

## Chapter 2 - Class Library

### ASNA.Monarch.WebDspf

---

#### ConditionalProperty Class

The **ConditionalProperty** is a base class that defines an array of conditional values (conditional indicator expressions). For a list of all members of this type, see ConditionalProperty Members.

ASNA.Monarch.WebDspf

##### **ASNA.Monarch.WebDspf.ConditionalProperty**

Derived classes

```
public class ConditionalProperty
```

#### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

#### Requirements

**Namespace:** ASNA.Monarch.WebDspf

**Assembly:** ASNA.Monarch.WebDspf

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

#### See Also

ASNA.Monarch.WebDspf Namespace

---

#### ConditionalProperty Constructor()

This method constructs a new instance of a **ConditionalProperty** object assigning default values for the properties.

```
BegConstructor ConditionalProperty Access(*Public)
```

#### Returns

A new instance of the **ConditionalProperty** object with default value property settings.

#### Requirements

**Namespace:** ASNA.Monarch.WebDspf

**Assembly:** ASNA.Monarch.WebDspf

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

#### See Also

ConditionalProperty Class

---

#### ConditionalProperty Constructor(string)

This method constructs a new instance of a **ConditionalProperty** object assigning an initial value to the PropString property.

```
BegConstructor ConditionalProperty Access(*Public)  
DeclSrParm propString Type(*string)
```

### Parameters

#### propString

String containing the initial value for the **PropString** property. This is an array of comma separated conditional values. For example, "07:03 & !99, 03:06, 99:05 ! 03"

### Returns

A new instance of the **ConditionalProperty** object with the **PropString** property set.

### Remarks

The composition of a conditional value expression is **value:condition**. For example (Red: 03 & !99), where **value** is "Red", and the **condition** is "03 & !99".

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ConditionalProperty Class

---

### Clone Method

Returns a copy of the **ConditionalProperty** object.

```
BegFunc Clone Access(*Public) Type(ConditionalProperty)
```

### Returns

An instance of **ConditionalProperty**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ConditionalProperty Class

---

### GetValidValues Method

Implementation of this method is provided in the derived classes.

```
BegFunc GetValidValues Access(*Public) Modifier(*MustOverride)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ConditionalProperty Class

---

## NewInstance Method

Implementation of this method is provided in the derived classes.

```
BegFunc NewInstance Access(*Public) Modifier(*MustOverride)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ConditionalProperty Class

---

## Reset Method

Restores the current **ConditionalProperty** object to default values.

```
BegFunc Reset Access(*Public) Type(Void)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ConditionalProperty Class

---

## ToString Method

Returns a string representation of the current **ConditionalProperty** object.

```
BegFunc ToString Access(*Public) Type(*String)
```

### Returns

String containing the current **ConditionalProperty** object.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ConditionalProperty Class

---

## Property Property

Returns or sets an array of ConditionalValue objects describing conditional indicator expression values for the **ConditionalProperty** object.

```
BegProp Property Access(*Public) Type(ConditionalValue)  
    BegGet; BegSet
```

## Property Value

ConditionalValue (get/set)

## Remarks

Property is an array of ConditionalValue objects that collectively describe the conditional indicator expressions for the ConditionalProperty object.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ConditionalProperty Class | ConditionalValue Class

---

## PropString Property

Returns a string containing the property string from the object represented by the **ConditionalProperty** object.

```
BegProp PropString Access(*Public) Type(*String)  
    BegGet
```

## Returns

String containing the property value string from the object represented by the **ConditionalProperty** object.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ConditionalProperty Class | ConditionalValue Class

---

## ConditionalValue Class

The **ConditionalValue** class defines a conditional indicator expression. For a list of all members of this type, see ConditionalValue Members.

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.ConditionalValue**

```
public class ConditionalValue
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

---

## ConditionalValue Constructor()

This method constructs a new instance of a **ConditionalValue** object.

```
BegConstructor ConditionalValue Access(*Public)
```

### Returns

A new instance of the **ConditionalValue** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ConditionalValue Class

---

## ConditionalValue Constructor(string, string)

This method constructs a new instance of a **ConditionalValue** object with the conditional indicator expression established.

```
BegConstructor ConditionalValue Access(*Public)
  DclSrParm condition Type(*string)
  DclSrParm val Type(*string)
```

### Parameters

#### condition

String containing the **condition** of the indicator expression as two operands. The first for when *val* evaluates true, the second when *val* evaluates false. Either can be \*NONE.

#### val

String containing the **value** for the condition expression.

## Returns

A new instance of the **ConditionalValue** object with conditional indicator expression established.

## Remarks

The composition of a conditional indicator expression is **value:condition**. An example using conditional indicators might be Red:03 & !99, where **value** is "Red", and the **condition** is "03 & !99".

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ConditionalValue Class

---

## ToString Method

Returns a string representation of the current **ConditionalValue** object.

```
BegFunc ToString Access(*Public) Type(*String)
```

## Returns

String containing the current **ConditionalValue** object.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ConditionalValue Class

---

## Condition Property

Returns or sets the *condition* portion of a conditional indicator expression.

```
BegProp Condition Access(*Public) Type(*String)  
BegGet; BegSet
```

## Property Value

String containing the **condition** of a conditional indicator expression as two operands. The first for when **value** evaluates true, the second when **value** evaluates false. Either can be \*NONE

## Remarks

The composition of a conditional indicator expression is **value:condition**. An example using conditional indicators might be Red:03 & !99, where the **Condition** property is "03 & !99".

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

---

## See Also

ConditionalValue Class

---

## Value Property

Returns or sets the *value* portion of a conditional indicator expression.

```
BegProp Value Access(*Public) Type(*String)  
    BegGet; BegSet
```

## Property Value

String containing the *value* portion of a conditional indicator expression.

## Remarks

The composition of a conditional indicator expression is **value:condition**. An example using conditional indicators might be Red:03 & !99, where the **Value** property is "Red".

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ConditionalValue Class

This page intentional left blank.

---

## DdsCharField Class

**DdsCharField** is a derived class that further defines a character field. It displays as a text box for user input.

For a list of all members of this type, see [DdsCharField Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.DdsCharField

```
public class DdsCharField: Inherits DdsInputField
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **DdsCharField** represents a user-input text box. The **DdsCharField** control has a **CssClass** called

Use the **Length** property to define the length of the character field.

Use the **Value** property to define a field value to the character field at runtime.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

---

## DdsCharField Constructor()

This method constructs a new instance of a **DdsCharField** object.

```
BegConstructor DdsCharField Access(*Public)
```

### Returns

A new instance of the **DdsCharField** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharField Class

---

## CheckedValue Property

For check box style, returns or sets the value to identify the box as being checked.

```
BegProp CheckedValue Access(*Public) Type(*String)  
BegGet; BegSet
```

### Property Value

String. The value to identify the box as being checked.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharField Class

---

### DefaultValue Property

Returns or sets a default value to display in the field.

```
BegProp DefaultValue Access(*Public) Type(*String)
    BegGet; BegSet
```

### Property Value

String. The default value to display in the field.

### Remarks

Enter a string containing the default value that you would like to display in the field. For Char Fields that are displayed as check boxes, **DefaultValue** defines whether the box is checked when the field is input-only.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharField Class

---

### InputStyles Property

Returns or sets the control style used to specify the field as TextBox, Multi-line textbox, Password, Checkbox.

```
BegProp InputStyles Access(*Public) Type(InputStyles)
    BegGet; BegSet
```

### Property Value

**InputStyles** reference for the style control for the field.

### Remarks

The default is **Textbox**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharField Class | InputStyles Enumeration

---

## Length Property

Returns or sets the total number of characters in the field.

```
BegProp Length Access(*Public) Type(*Integer)  
BegGet; BegSet
```

### Property Value

Integer. The total number of characters (length) of the field.

### Remarks

The **Length** of the field is used only when the **ValuesStyle** property is set to **Textbox**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharField Class | ValuesStyle Property

---

## Lower Property

Returns or sets a value indicating whether the user can type using all lower-case letters. The default is **True**.

```
BegProp Lower Access(*Public) Type(*Boolean)  
BegGet; BegSet
```

### Property Value

Boolean. **True**, which allows the user to input using lower-case letters; otherwise **False**.

### Remarks

The default is **True**, which allows the user to input using lower-case letters.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharField Class

---

## UncheckedValue Property

For check box style, returns or sets the value to identify the box as being unchecked.

```
BegProp UncheckedValue Access(*Public) Type(*String)  
BegGet; BegSet
```

### Property Value

String. The value to identify the box as being unchecked.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharField Class

---

## Value Property

Returns or sets the value of this field.

```
BegProp Value Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

String. The value for the field.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsCharFieldClass

---

## DdsConstant Class

The **DdsConstant** is a derived class that further defines a constant field.

For a list of all members of this type, see [DdsConstant Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.DdsConstant

```
public class DdsConstant Inherits DdsField
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

An instance of **DdsConstant** represents a constant field.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ASNA.Monarch.WebDspF Namespace

---

## DdsConstant Constructor()

This method constructs a new instance of a **DdsConstant** object.

```
BegConstructor DdsConstant Access(*Public)
```

## Returns

A new instance of the **DdsConstant** object with default values.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsConstant Class

---

## EncodedLiteral Method

This method takes a string value and encodes it to a valid XML element.

```
BegFunc EncodedLiteral Access(*Public) Type(*String)  
  DclSrParm text Type(*String)
```

## Parameters

### text

The text value to be encoded.

## Returns

String containing the *text* value with any replacement characters. <, >, and & become &lt;, &gt;, and &amp; respectively. Embedded spaces are not encoded with a replacement character.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsConstant Class

---

## EncodedNonBlank Method

This method takes a string value and encodes it to a valid XML element.

```
BegFunc EncodedNonBlank Access(*Public) Type(*String)
    DclSrParm text Type(*String)
```

## Parameters

### text

String. The value to be encoded.

### Returns

String containing the *text* value with any replacement characters. "<", ">", and "&" become "&lt;", "&gt;", and "&amp;" respectively. In addition, consecutive spaces (after the first) are replaced with "nbsp;".

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsConstant Class

---

## ResetText Method

Resets the Text property to default values.

```
BegFunc ResetText Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsConstant Class

---

## Text Property

Returns or sets a string containing either a constant to display, or the special value \*DATE, \*TIME, \*USER or \*SYSNAME.

```
BegProp Text Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. A constant or the system's special values \*DATE, \*TIME, \*USER or \*SYSNAME.

## Remarks

The default of **Constant** will display. You can change the text Constant to a constant value, such as Name:, or Address:, etc, or enter in one of the following special values.

Special Value	Description
*DATE	When you enter *DATE in the Text property, the contents of the field will display the current system's <b>date</b> , such as <b>1/27/2005</b> .
*TIME	When you enter *TIME in the Text property, the contents of the field will display the current system's <b>Time</b> , such as <b>4:00.00 PM</b> .
*USER	When you enter *USER in the Text property, the contents of the field will display the name of the current system's <b>User</b> , such as <b>Julie</b> .
*SYSNAME	When you enter *SYSNAME in the Text property, the contents of the field will display the current system's <b>Name</b> , such as <b>Juliexp</b> .

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsConstant Class

This page intentional left blank.

---

## DdsDecField Class

**DdsDecField** is a derived class that further defines a decimal field.

For a list of all members of this type, see [DdsDecField Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.DdsDecField

```
Public class DdsDecField Inherits DdsInputField
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **DdsDecField** represents a decimal field.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

---

## DdsDecField Constructor()

This method constructs a new instance of a **DdsDecField** object.

```
BegConstructor DdsDecField Access(*Public)
```

### Returns

A new instance of the **DdsDecField** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsDecField Class

---

## Decimals Property

Returns or sets the number of decimals in the field.

```
BegProp Decimals Access(*Public) Type(*Integer)  
BegGet; BegSet
```

### Property Value

Integer. The number of decimals in the field.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsDecField Class

## EditCode Property

Returns or sets a string containing an **edit code** to punctuate numeric fields according to the standard RPG edit code rules.

```
BegProp Edit code Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String containing an **edit code** to punctuate numeric fields as noted in the first column of the following table.

## Remarks

**Edit code** allows you to punctuate numeric fields, including \$ signs, commas, periods, minus sign, and floating minus according to the standard RPG edit code rules. Enter a valid **edit code** as listed in the table below that gives you the desired display. For example, entering a **J** in the **EditCode property** will display .00, giving you commas, decimal points, and a minus sign when necessary.

Edit Code	Commas	Dec Point	Sign for Negative Balance	Display
1	Yes	Yes	No Sign	.00 or 0
2	Yes	Yes	No Sign	Blanks
3	No	Yes	No Sign	.00 or 0
4	No	Yes	No Sign	Blanks
A	Yes	Yes	CR	.00 or 0
B	Yes	Yes	CR	Blanks
C	No	Yes	CR	.00 or 0
D	No	Yes	CR	Blanks
J	Yes	Yes	- (minus)	.00 or 0
K	Yes	Yes	- (minus)	Blanks
L	No	Yes	- (minus)	.00 or 0
M	No	Yes	- (minus)	Blanks
N	Yes	Yes	- (floating minus)	.00 or 0
O	Yes	Yes	- (floating minus)	Blanks

<b>P</b>	No	Yes	- (floating minus)	.00 or 0
<b>Q</b>	No	Yes	- (floating minus)	Blanks
<b>Y</b>		Yes	Date Edit	
<b>Z</b>		Yes	No sign	

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsDecField Class

## EditWord Property

Returns or sets a string containing an **edit word** to punctuate numeric fields according to the standard RPG rules for edit words.

```
BegProp EditWord Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. Returns or sets a string containing an **edit word** to punctuate numeric fields.

## Remarks

An **EditWord** allows you to directly specify:

- Blanks
- Output of the negative sign
- Suppression of unwanted zeroes
- Commas and decimal points
- Named Constants

At design-time, the **edit word** is not to be surrounded by an apostrophe.

### For example:

```
Use EditWord: 0bbb-bbb-bbbb
Not EditWord: '0bbb-bbb-bbbb'
```

The **Edit Word** is basically divided into three parts:

- The **Body** is the area for the digits transferred from the source data field to the output record. The body begins at the left most position of the edit word. The number of blanks, plus one zero or an asterisk in the edit word body must be equal to or greater than the number of digits of the source data field to be edited. The body ends with the rightmost character that can be replaced by a digit.
- The **Status** follows the last blank in the edit word, if used. It defines a space for a negative indicator, either a minus sign, or the two letters **CR**. The negative indicator will display only if the source data is negative. If the source data is positive, the status positions are replaced by blanks. Edit words without the CR or - indicators have no status positions. If more than one CR follows the last blank, the first CR is treated as a status, and the remaining CR's are treated as constants. The minus sign must be the last character in the edit word.

- The **Expansion** is entered after the **Status** and consists of a series of ampersands and constant characters. Ampersands are replaced by blank spaces in the output. Constants are output as is and follows the body, if status is not specified.

**The Body**

**Blank:** Blank is replaced with the character from the corresponding position of the Source Data field.

**Decimals and Commas:** Decimals and commas are in the same relative position in the edited output field as they are in the edit word unless they appear to the left of the first significant digit in the edit word. In that case, they are blanked out or replaced by an asterisk.

In the following examples, all the leading zeros will be suppressed (default) and the decimal point will not print unless there is a significant digit to its left.

Edit Word	Source Data	Appears as:
bbbbbbb	0000052	bbbbb52
bbbbbbb.bb	000000099	bbbbbbb99
bbbbbbb.bb	000000542	bbbbbb5.42

**Zeros:** The first zero in the **body** of the edit word is interpreted as an end-zero-suppression character. This zero is placed where zero suppression is to end. Subsequent zeros put into the edit word are treated as constants (see "Constants" below). Any leading zeros in the Source Data are suppressed up to and including the position of the end-zero-suppression character. Significant digits that would appear in the end-zero-suppression character position, or to the left of it, are output.

Edit Word	Source Data	Appears as:
bbb0bbbbbb	00000009	bbb000009
bbb0bbbbbb	098745	bbb098745
bbb0bbbbbb	098765432	bb98765432

If the leading zeros include, or extend to the right of, the end-zero-suppression character position, that position is replaced with a blank. This means that if you wish the same number of leading zeros to appear in the output as exist in the Source Data, the edit word body must be wider than the Source Data.

Edit Word	Source Data	Appears as:
0bbb	0541	b541
0bbbb	0541	b0541

**Constants** (including commas and decimal point) that are placed to the right of the end-zero-suppression character are output, even if there is no Source Data. Constants to the left of the end-zero-suppression character are only output if the Source Data has significant digits that would be placed to the left of these constants.

Edit Word	Source Data	Appears as:
bbbbbb0.bb	00000008	bbbbbb0.08
bbbbbb0.bb	00000000	bbbbbb0.bb
bbb,b0b.bb	00000045	bbbbbb0.45
bbb,b0b.bb	00000854	bbbbbb8.54
b0b,bbb.bb	00000854	bb0,008.54
b,b0b,bbb.bb	000008545	bbb0,008.54

**Asterisk:** The first asterisk in the body of an edit word also ends zero suppression. Subsequent asterisks put into the edit word are treated as constants (see "Constants" below). Any zeros in the edit word following this asterisk are also treated as constants. There can be only one end-zero-suppression character in an edit word, and that character is the first asterisk or the first zero in the edit word. If an asterisk is used as an end-zero-suppression character, all leading zeros that are suppressed are replaced with asterisks in the output. Otherwise, the asterisk suppresses leading zeros in the same way as described above for "Zeros".

Edit Word	Source Data	Appears as:
*bbbbbb.bb	000000854	*bbbbbb8.54
bbbbbb*b.bb	000000125	*****1.25
bbbbbb*b.bb**	000074524	****745.24**

Note that leading zeros appearing after the asterisk position are output as leading zeros. Only the suppressed leading zeros, including the one in the asterisk position, are replaced by asterisks.

**Currency Symbol:** A currency symbol followed directly by a first zero in the edit word (end-zero-suppression character) is said to float. All leading zeros are suppressed in the output and the currency symbol appears in the output immediately to the left of the most significant digit.

Edit Word	Source Data	Appears as:
bb,bbb,b\$0.bb	000000125	bbbbbbbbb\$1.25
bb,bbb,b\$0b.bb	000841458	bbb\$8,414.58

If the currency symbol is put into the first position of the edit word, then it will always appear in that position in the output. This is called a fixed currency symbol.

Edit Word	Source Data	Appears as:
\$b,bbb,bb0.bb	000854174	\$bbbb8,541.74
\$bb,bbb,0b0.bb	000000000	\$bbbbbb00.00
\$b,bbb*bb.bb	000745478	\$****7,454.78

A currency symbol anywhere else in the edit word and not immediately followed by a zero end-suppression-character is treated as a constant (see "Constants" below).

**Ampersand:** Creates a blank in the edited field. The examples below might be used to edit a telephone number. Note that the zero in the first position is required to print the constant AREA.

Edit Word	Source Data	Appears as:
0Area&bbb&No.&bbb-bbbb	9413784307	Area 941 No. 378-4307
0Area&[bbb]&No.&bbb-bbbb	9413784307	Area [941] No. 378-4307

**Constants:** All other characters entered into the body of the edit word re treated as constants. If the Source Data is such that the output places significant digits or leading zeros to the left of any constant, then that constant appears in the output. Otherwise, the constant is suppressed in the output. Commas and the decimal point follow the same rules as for constants. Notice in the examples below, that the presence of the end-zero-suppression character as well as the number of significant digits in the Source Data, influence the output of constants.

The following edit words could be used to print checks. Note that the second asterisk is treated as a constant, and that, in the third example, the constants preceding the first significant digit are not output.

Edit Word	Source Data	Appears as:
\$bbbbbb**Dollars&bb&Cts	000085474	\$****854*Dollars 74 Cts
\$bbbbbb**Dollars&bb&Cts	000000010	\$*****Dollars 10 Cts
\$bbbbbb&Dollars&bb&Cts	000000010	\$ 10 Cts

A date could be printed by using either edit word:

Edit Word	Source Data	Appears as:
bb/bb/bb	072396	7/23/96
0bb/bb/bb	072396	07/23/96

Note that any zeros or asterisks following the first occurrence of an edit word are treated as constants. The same is true for - and CR.

Edit Word	Source Data	Appears as:
bb0.bb000	08541	85.41000
bb*.bb000	08541	*85.41000

### The Status

The following characters have special meanings within the status:

- **Ampersand** - Causes a blank in the edited output field. An ampersand can not be placed in the edited output field.
- **CR or - symbol** - If the sign in the edited output is a plus (+), these positions are blanked out. If the sign in the edited output field is a minus (-), these positions remain undisturbed. The minus sign will only print when the field value is negative.

### The Expansion

The characters in this position are always displayed. This area can **not** contain blanks. If a blank is required in the edited output field, specify an ampersand in the body of the edit word.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsDecField Class

## Length Property

Returns or sets the length of the field.

```
BegProp Length Access(*Public) Type(*Integer)
    BegGet; BegSet
```

### Property Value

Integer. Returns or sets the length of the field.

### Remarks

The Length of the field is not used when the ValuesStyle property is set to anything other than **Textbox**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsDecField Class

## Type Property

Returns or sets the type of decimal field.

```
BegProp Type Access(*Public) Type(ASNA.Monarch.WebDspF.DecType)  
BegGet; BegSet
```

## Property Value

**ASNA.Monarch.WebDspF.DecType** for the type of decimal field.

## Remarks

The default value is **Zoned**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsDecField Class

---

## Value Property

Returns or sets the value for this field.

```
BegProp Value Access(*Public) Type(*String)  
BegGet; BegSet
```

## Property Value

String. The value for the field.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsDecField Class

---

## DdsField Class

**DdsField** is a base class that defines methods, properties and fields common to all 'field' controls in the derived classes.

For a list of all members of this type, see [DdsField Members](#).

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsField**

Derived classes

```
public class DdsField: Inherits System.Web.UI.WebControls.WebControl
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

---

## DdsField Constructor()

This method constructs a new instance of a **DdsField** object assigning default property values.

```
BegConstructor DdsField Access(*Public)
```

### Returns

A new instance of the **DdsField** object with default value property settings.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsField Class

---

## OnPreRender Method

This method raises the **PreRender** event that notifies the server control to perform any necessary prerending steps prior to saving new state and rendering content.

```
BegFunc OnPreRender Access(*Protected) Type(Void) Modifier(*Overrides)  
DclSrParm e Type(System.EventArgs)
```

### Parameters

e

**System.EventArgs** object that contains the event data.

## Remarks

This method is performed during the Prerender phase of the control execution lifecycle. The Prerender phase occurs when server control is about to render to its containing page object. During this phase, any field updates are made before the output is rendered.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsField Class

---

## ResetCheck Method

Resets the **Check** property to default values.

```
BegFunc ResetCheck Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsField Class

---

## ResetColor Method

Resets the **Color** property to default values.

```
BegFunc ResetColor Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsField Class

---

## ResetErrorMessage Method

Resets the **ErrorMessage** property to default values.

```
BegFunc ResetErrorMessage Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

---

## See Also

DdsField Class

---

## ResetErrorMessageId Method

Resets the **ErrorMessageId** property to default values.

```
BegFunc ResetErrorMessageId Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsField Class

---

## ResetVisibleCondition Method

Resets the **VisibleCondition** property to default values.

```
BegFunc ResetVisibleCondition Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsField Class

---

## Check Property

Returns or sets a new instance of a **CheckProperty** object for the field.

```
BegProp Check Access(*Public) Type(CheckProperty)  
BegGet; BegSet
```

## Property Value

**CheckProperty** object containing the conditional property values for the field (**Check Property Editor Dialog**).

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsField Class

## Color Property

Returns or sets a new instance of a `ColorProperty` object for the field.

```
BegProp Color Access(*Public) Type(ColorProperty)
  BegGet; BegSet
```

### Property Value

`ColorProperty` object containing the conditional property values for the field (**Color Property Editor Dialog**).

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsField Class

---

## Comment Property

Returns or sets a comment associated with the field.

```
BegProp Comment Access(*Public) Type(*String)
  BegGet; BegSet
```

### Property Value

`String` containing a comment for the field.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsField Class | ErrMsgProperty Class

---

## ErrorMessage Property

Returns or sets a new instance of an `ErrMsgProperty` object for the field.

```
BegProp ErrorMessage Access(*Public) Type(ErrMsgProperty)
  BegGet; BegSet
```

### Property Value

`ErrMsgProperty` object containing the conditional property values for the field (**ErrorMessage Property Editor**).

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

1DdsField Class | ErrMsgProperty Class

---

## ErrorMsgId Property

Returns or sets a new instance of an ErrMsgIdProperty object for the field.

```
BegProp ErrorMessageId Access(*Public) Type(ErrMsgIdProperty)  
    BegGet; BegSet
```

### Property Value

**ErrMsgIdProperty** object containing the conditional property values for the field (**ErrorMsgId Property Editor**).

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsField Class | ErrMsgIdProperty Class

---

## VisibleCondition Property

Returns or sets the RPG indicator expression that determines if the **DdsField** should be visible at any given time.

```
BegProp VisibleCondition Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

**String** containing the *RPG indicator* expression that determines if the **DdsField** should be visible at any given time.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsField Class

---

## IsDesignTime Field

Returns or sets a Boolean value indicating if the field is a design-time only field.

```
BegProp IsDesignTime Access(*Public) Type(*Boolean)  
    BegGet; BegSet
```

### Field Value

**Boolean** value **True** if the field is a design time only field; otherwise **false**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**See Also**

DdsField Class

## DdsFile Class

The **DdsFile** class is the container object responsible for enabling and processing function keys. It decides which function keys to draw on the form. When the user selects a function key, the appropriate indicators are turned on.

The **DdsFile** control is displayed as a key banner. If a function key is not enabled, the function keys in the banner are grayed out.

For a list of all members of this type, see DdsFile Members.

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.DdsFile

```
public class DdsFile Inherits System.Web.UI.Control
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

**DdsFile** has two important properties **AttnKeys** and **FuncKeys**

The **AttnKeys** property controls which keys the user can press, the message to be displayed, and the response indicators to turn on. The attention key settings are in a semi-colon(;) separated list of function key, 'text', and response: option indicators.

For example: F10 'Finished' 07 : !99;F4 'Skip' \*NONE : 09;

This property is mainly used to control processing within the application to control flow within the form since NO input data is transmitted from the browser.

The **FuncKeys** property contains all function key settings in a semi-colon(;) separated list of function key, 'text' and response: option indicators.

For example: F3 'Add' 07 : !99;F4 'Delete' \*NONE : 09;

This property is mainly used to control processing between applications since input data is transmitted from the browser.

## Requirements

**Namespace** : ASNA.Monarch.WebDspF

**Assembly**: ASNA.Monarch.WebDspF

**Platforms**: Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ASNA.Monarch.WebDspF Namespace

## DdsFile Constructor()

This method constructs a new instance of a **DdsFile** object assigning default property values.

```
BegConstructor DdsFile Access(*Public)
```

## Returns

A new instance of the **DdsFile** object with default value property settings.

## Requirements

**Namespace**: ASNA.Monarch.WebDspF

**Assembly**: ASNA.Monarch.WebDspF

**Platforms**: Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

---

## AddErrorMessage Method

This method adds an error message to the control.

```
BegFunc AddErrorMessage Access(*Public) Modifier(*Overrides) Type(*Boolean)  
  DclSrParm message Type(ASNA.Monarch.WebDspF.ErrorMessageInfo)  
  DclSrParm messageSource Type(*String)
```

## Property Values

### message

**ASNA.Monarch.WebDspF.ErrorMessageInfo** object to be added. This contains text of the message and information about replacement variables, and can include variable data that is being provided by the message sender.

### messageSource

**String** containing identification of the source of the message.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

---

## LoadPostData Method

Processes post back data for an ASP.NET server control.

```
BegFunc LoadPostData Access(*Public) Modifier(*Overrides) Type(*Boolean)  
  DclSrParm postDataKey Type(*String)  
  DclSrParm values Type(System.Collections.Specialized.NameValueCollection)
```

## Parameters

### postDataKey

String containing the identifier for the control.

### values

**System.Collections.Specialized.NameValueCollection** containing a collection of all incoming name values. Represents a sorted collection of associated System.String keys and System.String values that can be accessed either with the key or with the index.

## Returns

Returns **True** if the server control's state changed as a result of the postback; otherwise **False**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

---

## RaisePostDataChangedEvent Method

This method signals the server control object to notify the ASP.NET application that the state of the control has changed.

```
BegFunc RaisePostDataChangedEvent Access(*Public) Modifier(*Overrides)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class

---

## SetResponseIndicators Method

Sets the response indicator to handle subsequent action as a result of the feedback.

```
BegProp SetResponseIndicators Access(*Public) Type(*Integer)  
BegSet
```

### Returns

**Integer.** The indicator being set to handle subsequent action as a result of the feedback.

### Remarks

The response indicators are used to pass information from the system to an application program when an input request completes. These indicators can be used to determine which function keys were pressed or whether data was changed. Response indicators can be specified at the file, record format, and field levels

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class

---

## AttnID Property

Gets the identification of the function key used to focus attention on this control.

```
BegProp AttnID Access(*Public) Type(*Integer)  
BegGet
```

### Property Value

**Integer.** The identification of the function key used to focus attention on this control.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

---

## AttnKeys Property

Returns or sets which keys the user can press, the message to be displayed, and the response indicators to turn on. No input data is transmitted from the Browser.

```
BegProp AttnKeys Access(*Public) Type(ASNA.Monarch.WebDspF.AidProperty)
BegGet; BegSet
```

## Property Value

**ASNA.Monarch.WebDspF.AidProperty** containing the attention key settings in a semi-colon(;) separated list.

## Remarks

The **AttnKeys** property contains all attention key settings in a semi-colon(;) separated list of function key, 'text' and response:option indicators.

For example: `F10 'Finished' 07 : !99;F4 'Skip' *NONE : 09;`

This property is mainly used to control processing within the application to control flow within the form since NO input data is transmitted from the browser. Use the **FuncKeys** Property to specify function keys that transfer input to the Browser.

To set this property at design-time, click on the right of the **AttnKeys** property and the **AidProperty Dialog** will display. Enter the Text and resulting option indicators in the row corresponding to the key to apply to. Any unused Function keys are grayed out when displayed in the key banner.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class | **FuncKeys** Property

---

## BackColor Property

Returns or sets the color of the background color of the key banner.

```
BegProp BackColor Access(*Public) Type(System.Drawing.Color)
BegGet; BegSet
```

## Property Value

**System.Drawing.Color** that represents the background color of the key banner.

## Remarks

The default is **Transparent**.

To set the **BackColor** property at design time, click the arrow to the right to **BackColor** property and the System Color dialog will display as shown below. Select the color by first selecting the **Custom**, **Web**, or **System** tabs and then clicking on the background color desired.



## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

---

## BannerStyle Property

Returns or sets how the function key banner on the page will display. **Horizontal** is the default.

```
BegProp BannerStyle Access(*Public) Type(ASNA.Monarch.WebDspF.BannerStyles)  
BegGet; BegSet
```

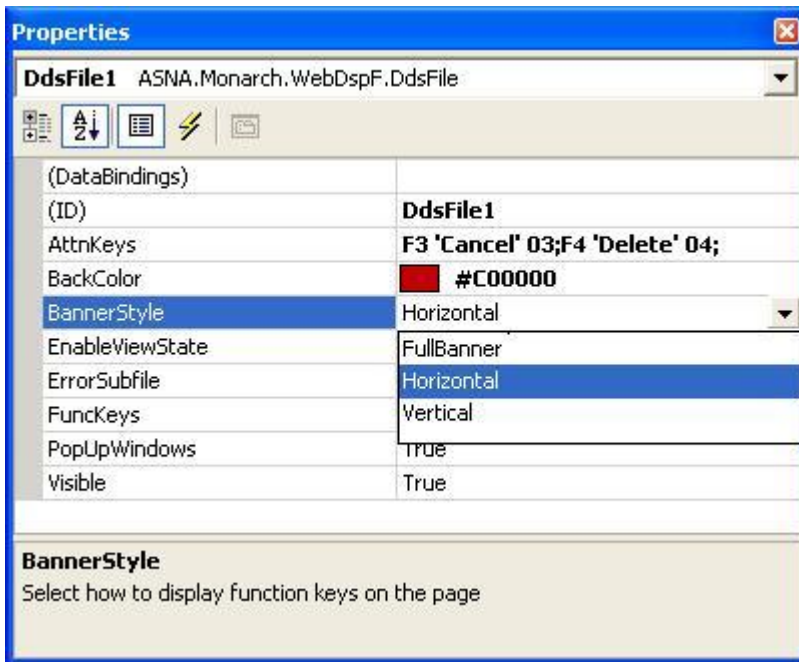
## Property Value

**ASNA.Monarch.WebDspF.BannerStyles** that represents how the function keys banner is to display on the page.

## Remarks

The **BannerStyle** property determines whether the keys will display as either a full banner, horizontal or vertical. The default is **horizontal**.

To set this property at design-time, click to the right of the **BannerStyle** property and select the style from the drop-down list as show below.



### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class

---

## CursorField Property

Returns the name of the field within the record format at the current cursor position.

```
BegProp CursorField Access(*Public) Type(*String)
    BegGet
```

### Returns

String containing the field name at the current cursor position.

### Remarks

Use the **CursorRecord** property to return the name of the record format for the field at the cursor's location, and the **CursorLocation** property to determine the cursor's position within the current field.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class

---

## CursorLocation Property

Returns the position of the cursor within the current field.

```
BegProp CursorLocation Access(*Public) Type(*Integer)  
BegGet
```

### Returns

Integer containing the current cursor position within the field.

### Remarks

Use the **CursorRecord** property to return the name of the record format for the field at the cursor's location, and the **CursorField** property to return the field name at the cursor's current position.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class

---

## CursorRecord Property

Returns the name of the record format for the field at the current cursor position.

```
BegProp CursorRecord Access(*Public) Type(*String)  
BegGet
```

### Returns

String containing the record format for the field at the current cursor position.

### Remarks

Use the **CursorField** property to return the field name at the cursor's current position and the **CursorLocation** property to determine the cursor's position within the current field

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class

---

## ErrorSubfile Property

Returns or sets a Boolean value indicating whether messages are to be displayed in a subfile-style. The default is **False**.

```
BegProp ErrorSubfile Access(*Public) Type(*Boolean)  
BegGet; BegSet
```

### Property Value

Boolean. **True** indicates messages are to be displayed in a subfile-style; otherwise **False**.

### Remarks

The default is **False**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class

---

## FuncKeys Property

Returns or creates an instance of **ASNA.Monarch.WebDspF.AidProperty** specifying which keys the user can press, the message to be displayed, and the response indicators to turn. Input data is transmitted from the Browser.

```
BegProp FuncKeys Access(*Public) Type(ASNA.Monarch.WebDspF.AidProperty)  
BegGet; BegSet
```

### Property Value

**ASNA.Monarch.WebDspF.AidProperty** containing the function key settings in a semi-colon(;) separated list.

### Remarks

The **FuncKeys** property contains all function key settings in a semi-colon(;) separated list of function key, 'text' and response: option indicators.

For example: F3 'Add' 07 : !99;F4 'Delete' \*NONE : 09;

This property is mainly used to control processing between applications since input data is transmitted from the browser. Use the **AttnKeys** Property to specify function keys that do not transfer input to the Browser.

To set this property at design-time, click on the right of the **FuncKeys** property and the **AidProperty Dialog** will display. Enter the Text and resulting option indicators in the row corresponding to the key to apply to.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsFile Class | AttnKeys Property

---

## PopUpWindows Property

Returns or sets a Boolean value indicating if record formats marked as Window (see [Window\\_property](#)) will render as pop-ups in the browser. The default is **True**.

```
BegProp PopUpWindows Access(*Public) Type(*Boolean)  
BegGet; BegSet
```

### Property Value

Boolean. **True** indicates record formats marked Window (\*True) will render as pop-ups in the browser; otherwise **False**.

---

## Remarks

The **PopUpWindows** property has no effect if none of the record formats are marked as Window.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

---

## ResultingIndicators Property

Gets the indicators used to pass control to the message handler for subsequent action as a result of the feedback.

```
BegProp ResultingIndicators Access(*Public) Type(*String)
    BegGet
```

## Returns

**String.** The indicators used to pass control to the message handler for subsequent action as a result of the feedback.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

---

## InputDataAvailable Field

Returns or sets a Boolean value indicating if input data is available to the control.

```
BegProp InputDataAvailable Access(*Public) Type(*Boolean)
    BegGet; BegSet
```

## Field Value

**Boolean** value **True** if input data is available; otherwise **false**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsFile Class

This page intentionally left blank

## DdsInputField Class

**DdsInputField** is a derived class that further defines properties and methods common to display file fields. In addition, **DdsInputField** is also a base class for **DdsCharField**, **DdsDecField** and **DdsTimeStamp**.

For a list of all members of this type, see [DdsInputField Members](#).

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsField**

**ASNA.Monarch.WebDspF.DdsInputField**

```
Public class DdsInputField : Inherits ASNA.Monarch.WebDspF.DdsField
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

In addition to the methods, properties and fields of **ASNA.Monarch.WDspFile.DdsField**, there are additional methods and properties specific to input fields.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

---

## DdsInputField Constructor()

This method constructs a new instance of a **DdsInputField** object.

```
BegConstructor DdsInputField Access(*Public)
```

### Returns

A new instance of the **DdsInputField** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

---

## LoadPostData Method

Processes post back data for an ASP.NET server control.

```
BegFunc LoadPostData Access(*Public) Modifier(*Overrides) Type(*Boolean)
  DclSrParm postDataKey Type(*String)
  DclSrParm values Type(System.Collections.Specialized.NameValueCollection)
```

### Parameters

**postDataKey**

String containing the identifier for the control.

**values**

**System.Collections.Specialized.NameValueCollection** containing a collection of all incoming name values. Represents a sorted collection of associated System.String keys and System.String values that can be accessed either with the key or with the index.

### Returns

Returns **True** if the server control's state changed as a result of the postback; otherwise **False**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

---

### RaisePostDataChangedEvent Method

This method signals the server control object to notify the ASP.NET application that the state of the control has changed.

```
BegFunc RaisePostDataChangedEvent Access(*Public) Modifier(*Overrides)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

---

### RenderBeginTag Method

This method renders the HTML opening tag of the control into the specified *writer*.

```
BegFunc RenderBeginTag Access(*Public) Modifier(*Overrides)  
DeclSrParm writer Type(System.Web.UI.HtmlTextWriter)
```

### Parameters

**writer**

A **System.Web.UI.HtmlTextWriter** that represents the output stream to render HTML content on the client.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

---

## RenderEndTag Method

This method renders the HTML closing tag of the control into the specified *writer*.

```
BegFunc RenderEndTag Access(*Public) Modifier(*Overrides)  
DclSrParm writer Type(System.Web.UI.HtmlTextWriter)
```

### Parameters

**writer**

A **System.Web.UI.HtmlTextWriter** that represents the output stream to render HTML content on the client.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputFieldClass

---

## ResetMessageId Method

This method resets the **MessageId** property.

```
BegFunc ResetMessageId Access(*Public)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

---

## Alias Property

Returns or sets an alternate field name for the field.

```
BegProp Alias Access(*Public) Type(*String)  
BegGet; BegSet
```

### Property Value

String. The alternate name for the field. This value must be different from all other alternative names and from all DDS field names in the record format

### Remarks

When the program is compiled, the alternative name is brought into the program instead of the DDS field name.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

## ChangeInd Property

Returns or sets the response indicator to set 'on' when this field changes.

```
BegProp ChangeInd Access(*Public) Type(*Integer)  
BegGet; BegSet
```

### Property Value

Integer. The response indicator specifying that the field changed.

### Remarks

The default value is 0. Use this property to set on the specified response indicator for an input operation when the user keys into this field. When the record is displayed again with an error message, the response indicator that have been set on by typing into fields remain on until all validity checks succeed and the record is passed to your program.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

---

## MsgId Property

Returns or sets a reference to an **MsgIdProperty** object for the message ID to be associated with this field.

```
BegProp MsgId Access(*Public) Type(MsgIdProperty)  
BegGet; BegSet
```

### Property Value

**ASNA.Monarch.WebDspF.MsgIdProperty** object for the message ID to be associated with this field.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsInputField Class

---

## PositionCursor Property

Returns or sets a string containing the position cursor attribute value for the

```
BegProp PositionCursor Access(*Public) Type(*String)  
BegGet; BegSet
```

### Property Value

String containing the position cursor attribute for the field.

## Remarks

Set this property to position the cursor to the first character position of the field you are defining. You can specify this property for several fields, and the cursor will be positioned at the first display field with this attribute. (**DspAtr(PC)**)

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsInputField Class

## Protect Property

Return or sets a string indicating if the field is protected. Set **\*True** indicates the field is Output-Only (read-only), otherwise **\*False**.

```
BegProp Protect Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. **\*True** if the field is Output-Only (read-only), otherwise **\*False**.

## Remarks

Setting this property has not affect for Hidden fields. (**DspAtr(PR)**)

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsInputField Class

## Text Property

Returns or sets the text contained in the control for this field.

```
BegProp Text Access(*Protected) Type(*String) Modifier(*NotOverridable)
    BegGet; BegSet
```

## Property Value

String containing the text for the control.

## Remarks

Used only for **ValuesStyle.Textbox** or **Checkbox**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsInputField Class

---

## Usage Property

Returns or set how the field is used, e.g.: output, input, both (input and output), or hidden.

```
BegProp Usage Access(*Public) Type(FieldUsages)
    BegGet; BegSet
```

## Property Value

**FieldUsages** reference for how the field is used.

## Remarks

The default is **both** (input and output).

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsInputField Class | FieldUsages Enumeration

---

## Values Property

Returns or sets the list of valid values that the user can input into the field; separated by blanks.

```
BegProp Value Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. The list of valid values for the field; separated by blanks.

## Remarks

The **Values** property allows you to display for the user valid values that the input field can accept in the application; separated by spaces. However, if the **Values** property is left blank, which is the default, the user can enter anything into the field.

For example, if the application is looking for an input value of A, B, C or D, you would list the values **A B C D** in the **Values** property.

However, unless the user doing the input knows that those Values stand for, those values can be confusing. If using Values, you may want to also set the [ValuesText](#) property, which allows you to specify one word per **Value** entered in the **Values** property; separated by spaces.

For example, using **A B C D** in the **Values** property, you could enter the text **Add Browse Change Delete** in the **ValueText** property. The user would then see the following in the field:

A = Add

B = Browse

C = Change

D = Delete

For the user to see these **Values**, the [ValuesStyle](#) property must be set to either **DropdownValues** or **DropdownBoth** (also displays values in **ValueText** property).

**Note:** If there are no entries in the **Values** property, then regardless if there are entries for the **ValuesStyle** and **ValuesText** properties, they will be disregarded and the field will display as a **TextBox**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsCharFieldClass | DdsCharField Class Members | ASNA.Monarch.WebDspF Namespace | ValuesText Property | ValuesStyle Property

## ValuesStyle Property

Returns or sets the list of valid values that the user can input into the field; separated by blanks.

```
BegProp Value Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. The list of valid values for the field; separated by blanks.

## Remarks

The **Values** property allows you to display for the user valid values that the input field can accept in the application; separated by spaces. However, if the **Values** property is left blank, which is the default, the user can enter anything into the field.

For example, if the application is looking for an input value of A, B, C or D, you would list the values **A B C D** in the **Values** property.

However, unless the user doing the input knows that those Values stand for, those values can be confusing. If using Values, you may want to also set the [ValuesText](#) property, which allows you to specify one word per **Value** entered in the **Values** property; separated by spaces.

For example, using **A B C D** in the **Values** property, you could enter the text **Add Browse Change Delete** in the **ValueText** property. The user would then see the following in the field:

A = Add

B = Browse

C = Change

D = Delete

For the user to see these **Values**, the [ValuesStyle](#) property must be set to either **DropdownValues** or **DropdownBoth** (also displays values in **ValueText** property).

**Note:** If there are no entries in the **Values** property, then regardless if there are entries for the **ValuesStyle** and **ValuesText** properties, they will be disregarded and the field will display as a **TextBox**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsCharFieldClass | DdsCharField Class Members | ASNA.Monarch.WebDspF Namespace | ValuesText Property | ValuesStyle Property

## ValuesText Property

Returns or sets the list of text to be shown when using the **ValuesStyle** property is set to **DropDownText** or **DropDownBoth**.

```
BegProp Value Access(*Public) Type(*String)
    BegGet; BegSet
```

### Property Value

String. The list of valid values for the field; separated by blanks.

### Remarks

The **ValuesText** property allows you to display for the user valid values that the input field can accept in the application; separated by spaces. However, if the Values property is left blank, which is the default, the user can enter anything into the field.

For example, if the application is looking for an A, B, C or D, you would list the values **A B C D** in the **Values** property.

However, unless the user doing the input knows that those values stand for, those values can be confusing. If using Values, you may want to also set the **ValueText** property, which allows you to specify one word per **Value** entered in the **Values** property; separated by spaces.

For example, using **A B C D** in the **Values** property, you could enter the text **Add Browse Change Delete** in the **ValueText** property. The user would then see the following in the field:

A = Add

B = Browse

C = Change

D = Delete

### Requirements

**Namespace:** [ASNA.Monarch.WebDspF](#)

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsCharFieldClass](#) | [DdsCharField Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#) | [ValuesStyle Property](#) | [Values Property](#)

---

## VirtualRowCol Property

Returns or sets the row and column that this field is reported on in the Display file.

```
BegProp VirtualRowCol Access(*Public) Type(*String)
    BegGet; BegSet
```

### Property Value

String containing the row and column this field is reported on in the Display file; such as **7,27**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**See Also**

[DdsInputFieldClass](#) | [DdsInputField Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

This page intentionally left blank.

---

## DdsLink Class

The **DdsLink** class defines a hyperlink field.

For a list of all members of this type, see [DdsLink Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.DdsLink

```
Public class DdsLink Inherits System.Web.UI.WebControls.WebControl
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **DdsLink** represents a hyperlink field.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ASNA.Monarch.WebDspF Namespace](#) | [DdsLink Members](#)

---

## DdsLink Constructor()

This method constructs a new instance of a **DdsLink** object.

```
BegConstructor DdsLink Access(*Public)
```

### Returns

A new instance of the **DdsLink** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsLink Class](#) | [DdsLink Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ResetVisibleCondition Method

This method resets the **VisibleCondition** property.

```
BegFunc ResetVisibleCondition Access(*Public)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsLink Class](#) | [DdsLink Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### TextFieldLength Property

Returns or sets the total number of characters in the **TextFieldName**.

```
BegProp TextFieldLength Access(*Public) Type(*Integer)
BegGet
```

### Property Value

Integer. The number of characters that are in the **TextFieldName**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsLinkClass](#) | [DdsLink Class Members](#) | [TextFieldName Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### TextFieldName Property

Returns or sets the field name for the link.

```
BegProp TextFieldName Access(*Public) Type(*String)
BegGet; BegSet
```

### Property Value

String. The field name.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsLinkClass](#) | [DdsLink Class Members](#) | [TextFieldName Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### TextValue Property

Sets the text to display on the link field.

```
BegProp TextValue Access(*Public) Type(*String)
BegSet
```

### Property Value

String. The text to display on the link.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

---

## See Also

[DdsLinkClass](#) | [DdsLink Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## UrlFieldName Property

Returns or sets the number of characters that are in the **UrlFieldName**.

```
BegProp UrlFieldName Access(*Public) Type(*Integer)
    BegGet; BegSet
```

## Property Value

Integer. The total number of characters in the **UrlFieldName**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsLinkClass](#) | [DdsLink Class Members](#) | [UrlFieldName Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## UrlFieldName Property

Returns or sets the url field name containing the link.

```
BegProp UrlFieldName Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. The url field name for the link.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsLinkClass](#) | [DdsLink Class Members](#) | [UrlFieldName Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## UrlValue Property

Sets the underlying hyperlink value.

```
BegProp UrlValue Access(*Public) Type(*String)
    BegSet
```

## Property Value

String. The underlying hyperlink value.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsLinkClass | DdsLink Class Members | ASNA.Monarch.WebDspF Namespace

---

### VirtualRowCol Property

Returns or sets the fields row/col position in the Display file.

```
BegProp VirtualRowCol Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

String containing the row/col position in the Display file.

### Remarks

Enter a string containing the row/col positions for this field within the Display file; such as **7,27**. The row and column in the display file determines the order of HTML tags in the HTML document that is created.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsLinkClass | DdsLink Class Members | ASNA.Monarch.WebDspF Namespace

---

### VisibleCondition Property

Returns or sets a set of conditional indicators that determine the conditions under which the field is visible and rendered.

```
BegProp VisibleCondition Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

String. A set of conditional indicators that determine the conditions under which the field is visible and rendered.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsLinkClass | DdsLink Class Members | ASNA.Monarch.WebDspF Namespace

## DdsRecord Class

The **DdsRecord** base class that is a container for the record fields and constants. The record is displayed as a panel with a grid layout.

For a list of all members of this type, see [DdsRecord Members](#).

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsRecord**

**Derived classes**

```
Public class DdsRecord: Inherits System.Web.UI.WebControls.Panel
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **DdsRecord** represents a container for the fields and constants of the record. This class also inherits the methods and properties of **System.Web.UI.WebControls.Panel**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ASNA.Monarch.WebDspF Namespace](#) | [DdsRecord Members](#)

## DdsRecord Constructor()

This method constructs a new instance of a **DdsRecord** object.

```
BegConstructor DdsRecord Access(*Public)
```

### Returns

A new instance of the **DdsRecord** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## CollectResponseIndicators Method

This method returns a string array containing the response indicators for the

```
BegFunc CollectResponseIndicators Access(*Public)
DclSrParm responseIndicators Type(System.Collections.ArrayList) Rank(1)
```

### Parameters

**responseIndicators**

**System.Collections.ArrayList** that represents an array of response indicators for the record.

## Returns

String array containing the response indicators for the record.

## Remarks

The response indicators are used to pass information from the system to an application program when an input request completes. These indicators can be used to determine which function keys were pressed or whether data was changed. Response indicators can be specified at the file, record format, and field levels.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsRecord Class | DdsRecord Class Members | ASNA.Monarch.WebDspF Namespace

---

## GetConditionedValue Method(ASNA.Monarch.WebDspF.ConditionalValue)

This method returns a string array containing the conditional values for the conditions to return.

```
BegFunc GetConditionedValue Access(*Public) Type(*String)
DclSrParm possibleValues Type(ASNA.Monarch.WebDspF.ConditionalValue) Rank(1)
```

## Parameters

### possibleValues

An array of **ASNA.Monarch.WebDspF.ConditionalValue** objects that represent the conditions to return.

## Returns

String array containing the conditional indicator expressions for the conditions to return. The composition of a conditional indicator expression is **value:condition**. An example using conditional indicators might be Red:03 & !99, where **value** is "Red", and the **condition** is "03 & !99".

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsRecord Class | DdsRecord Class Members | ConditionalValue Class | ASNA.Monarch.WebDspF Namespace

---

## GetConditionedValue Method(string, ASNA.Monarch.WebDspF.ConditionalValue)

This method returns a string array containing the conditional values for a specific field.

```
BegFunc GetConditionedValue Access(*Public) Type(*String)
DclSrParm fieldID Type(*string)
DclSrParm possibleValues Type(ASNA.Monarch.WebDspF.ConditionalValue) Rank(1)
```

## Parameters

### fieldID

String containing the ID for the field to get the conditional values for.

### possibleValues

An array of **ASNA.Monarch.WebDspF.ConditionalValue** objects that represents the conditions to return.

## Returns

String array containing the conditional indicator expressions for the specific field. The composition of a conditional indicator expression is **value:condition**. An example using conditional indicators might be Red:03 & !99, where **value** is "Red", and the **condition** is "03 & !99".

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ConditionalValue Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## GetDdsFile Method

This method returns an instance of a **DdsFile** object for the

```
BegFunc GetDdsFile Access(*Public) Type(DdsFile)
```

## Returns

An instance of an **DdsFile** object.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecord Class Members](#) | [ChangeInd Property](#) | [DdsFile Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## GetResultIndicator Method

This method returns the integer value for the result indicator specified.

```
BegFunc GetResultIndicator Access(*Public) Type(*Integer)  
  DclSrParm indValue Type(*String)
```

## Parameters

**indValue**

**String** that contains the indicator value.

## Returns

Integer value containing 0 (False) or 1 (True).

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecord Class Members](#) | [ChangeInd Property](#) | [DdsFile Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

## IsAidKeyInEffect Method(ASNA.Monarch.WebDspF.ConditionalValue, integer)

This overloaded method returns a Boolean value indicating if the **Aidkey** Property Dialog is in effect. If in effect, the resulting indicator is returned

```
BegFunc IsAidKeyInEffect Access(*Public)
  DclSrParm indExpression Type(ASNA.Monarch.WebDspF.ConditionalValue) Rank(1)
  DclSrParm resultingIndicator Type(*Integer)
```

### Parameters

#### indExpression

An array of **ASNA.Monarch.WebDspF.ConditionalValue** objects that represents the indicator expression.

#### resultingIndicator

Integer. The indicator set 'on' as a result of the indicator expression.

### Returns

**True** if the method determines the indicator expression represents an **AidKey** Property has been set and the resultingIndicator will contain the indicator value set by the evaluation of the indicator expression; otherwise **False**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ConditionalValue Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

## IsAidKeyInEffect Method(ASNA.Monarch.WebDspF.AidKey, integer)

This overloaded method returns a Boolean value indicating if the Aidkey Property Dialog is in effect. If in effect, the resulting indicator and text description of the key are returned.

```
BegFunc IsAidKeyInEffect Access(*Public)
  DclSrParm key Type(ASNA.Monarch.WebDspF.AidKey) Rank(1)
  DclSrParm resultingIndicator Type(*Integer)
  DclSrParm text Type(*String)
```

### Parameters

#### key

An **ASNA.Monarch.WebDspF.AidKey** object that represents the function key.

#### resultingIndicator

Integer. The indicator to set 'on' if the key is pressed.

#### text

String. The text description of the key.

### Returns

**True** if the method determines the indicator expression represents an **AidKey** Property has been set; otherwise **False**. If **True**, the *resultingIndicator* will contain the indicator, and *text* will contain the text description of the function key.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsRecord Class | DdsRecord Class Members | ASNA.Monarch.WebDspF Namespace

---

## IsNotFalse Method(string)

This method returns **True** if the condition is not false.

```
BegFunc IsNotFalse Access(*Public)
DclSrParm condition Type(*string)
```

### Parameters

#### condition

String containing the condition to evaluate.

### Returns

**True** if the *condition* evaluates not true; otherwise **False**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsRecord Class | DdsRecord Class Members | ConditionalValue Class | ASNA.Monarch.WebDspF Namespace

---

## IsNotFalse Method(string, string)

This method returns **True** if the condition for the fieldID is not false.

```
BegFunc IsNotFalse Access(*Public)
DclSrParm fieldID Type(*string)
DclSrParm condition Type(*string)
```

### Parameters

#### fieldID

String containing the fieldID to evaluate.

#### condition

String containing the condition to evaluate for the *fieldID*.

### Returns

**True** if the *condition* for the *fieldID* is not false; otherwise **False**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsRecord Class | DdsRecord Class Members | ConditionalValue Class | ASNA.Monarch.WebDspF Namespace

## IsTrue Method(string)

This method returns **True** if the condition evaluates true.

```
BegFunc IsTrue Access(*Public)Type(*Boolean)  
DclSrParm condition Type(*string)
```

### Parameters

#### condition

String containing the condition to evaluate.

### Returns

**True** if the *condition* evaluates true; otherwise **False**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ConditionalValue Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## IsTrue Method(string, string)

This method returns **True** if the condition for the fieldID is true.

```
BegFunc IsTrue Access(*Public)Type(*Boolean)  
DclSrParm fieldID Type(*string)  
DclSrParm condition Type(*string)
```

### Parameters

#### fieldID

String containing the ID for the field to evaluate.

#### condition

String containing the condition to evaluate for the fieldID.

### Returns

**True** if the *condition* for the *fieldID* is true; otherwise **False**

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ConditionalValue Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## OnPreRender Method

This method raises the **PreRender** event that notifies the server control to perform any necessary prerending steps prior to saving new state and rendering content.

```
BegFunc OnPreRender Access(*Protected) Type(Void) Modifier(*Overrides)  
DclSrParm e Type(System.EventArgs)
```

---

## Parameters

e

**System.EventArgs** object that contains the event data.

## Remarks

This method is performed during the Prerender phase of the control execution lifecycle. The **Prerender** phase occurs when server control is about to render to its containing page object. During this phase, any field updates are made before the output is rendered.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsRecord Class | DdsRecord Class Members | Control Execution Lifecycle | ASNA.Monarch.WebDspF Namespace

---

## SetDataChanged Method

This method sets the result indicator to specified that data has changed within the

```
BegFunc SetDataChanged Access(*Public)
DclSrParm resultInd Type(*Integer)
```

## Parameters

**resultInd**

Integer that represents the result indicator to turn on to specify that data has changed.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsRecord Class: Members, ChangeInd property | ASNA.Monarch.WebDspF Namespace

---

## SetOptionIndicators Method

This method sets the option indicators for the

```
BegFunc SetOptionIndicators Access(*Public)
DclSrParm StarINstring Type(*String)
```

## Parameters

**StarINstring**

**String** that represents an \*INxx.

## Remarks

The option indicators are used to pass information from an application program to the system. These indicators can be used to control the processing for the record.

In Visual RPG, an indicator array \*IN is automatically defined for each program. Individual elements can be accessed using the array element syntax ([ ] or by using the predefined names \*IN01, \*IN02, ... etc. The \*IN array is 99 long.

\*INLR is another special predefined indicator that is used to determine the program end, if the program should be unloaded.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## AttnKeys Property

Returns or creates an instance of **ASNA.Monarch.WebDspF.AidProperty** which contains the keys the user can press, the message to be displayed, and the response indicators to turn on. No input data is transmitted from the Browser.

```
BegProp AttnKeys Access(*Public) Type(ASNA.Monarch.WebDspF.AidProperty)
BegGet: BegSet
```

## Property Value

**ASNA.Monarch.WebDspF.AidProperty** containing the attention key settings in a semi-colon(;) separated list.

## Remarks

The **AttnKeys** property contains all attention key settings in a semi-colon(;) separated list of function key, 'text' and response:option indicators.

For example: `F10 'Finished' 07 : !99;F4 'Skip' *NONE : 09;`

This property is mainly used to control processing within the application to control flow within the form since NO input data is transmitted from the browser. Use the **FuncKeys Property** to specify function keys that transfer input to the Browser.

To set this property at design-time, click on the right of the **AttnKeys** property and the **AidProperty Dialog** will display. Enter the Text and resulting option indicators in the row corresponding to the key to apply to. Any unused Function keys are grayed out when displayed in the key banner.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecordClass](#) | [DdsRecord Class Members](#) | [FuncKeys Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ChangeInd Property

Returns or sets the response indicator to set 'on' when this record changes.

```
BegProp ChangeInd Access(*Public) Type(*Integer)
BegGet; BegSet
```

## Property Value

Integer. Specifies which indicator to set on when the **DdsRecord** changes.

## Remarks

The default value is 0.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsRecordClass | DdsRecord Class Members | ASNA.Monarch.WebDspF Namespace

## EraseFormats Property

Returns or creates an instance of **ASNA.Monarch.WebDspF.EraseProperty** specifying which records are to be erased from the display when this record is written.

```
BegProp EraseFormats Access(*Public) Type(ASNA.Monarch.WebDspF.EraseProperty)
BegGet; BegSet
```

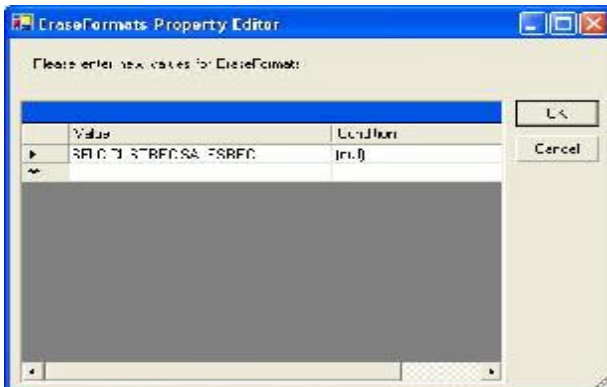
## Property Value

**ASNA.Monarch.WebDspF.EraseProperty** object containing the record formats to be erased from the display.

## Remarks

The **EraseFormats** property allows you to specify which records you want to be erased from the display when this record is written. The default is **\*ALL**, which indicates to remove all records.

To set the **EraseFormats** property at design-time, click on the far right side of the **EraseFormats** property and the **EraseFormats** Property Editor will display as shown below.



## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsRecordClass | DdsRecord Class Members | ASNA.Monarch.WebDspF Namespace

## FuncKeys Property

Returns or creates an instance of **ASNA.Monarch.WebDspF.AidProperty** specifying which keys the user can press, the message to be displayed, and the response indicators to turn on. Input data is transmitted from the Browser.

```
BegProp FuncKeys Access(*Public) Type(ASNA.Monarch.WebDspF.AidProperty)  
    BegGet; BegSet
```

### Property Value

**ASNA.Monarch.WebDspF.AidProperty** containing the function key settings in a semi-colon(;) separated list.

### Remarks

The **FuncKeys** property contains all function key settings in a semi-colon(;) separated list of function key, 'text' and response: option indicators.

For example: F3 'Add' 07 : !99;F4 'Delete' \*NONE : 09;

This property is mainly used to control processing between applications since input data is transmitted from the browser. Use the **AttnKeys Property** to specify function keys that do not transfer input to the Browser.

To set this property at design-time, click on the right of the **FuncKeys** property and the **AidProperty Dialog** will display. Enter the Text and resulting option indicators in the row corresponding to the key to apply to. You can not enter the

When the user pressing the function key defined, the record containing the changed fields is returned to the program. If a response indicator is also specified, the response indicator is set on and also passed to the program with the input data.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecordClass](#) | [DdsRecord Class Members](#) | [AttnKeys Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ReturnData Property

Returns or sets the Boolean value specifying whether or not to return the same data that was returned on the previous input operation sent to this record format (RTNDATA). The default is **False**.

```
BegProp ReturnData Access(*Public) Type(*Boolean)  
    BegGet; BegSet
```

### Property Value

Boolean. **True** to return the same data that was returned on the previous input operation sent to this record format; otherwise **False**.

### Remarks

The default is **False**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecordClass](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## Window Property

Returns or sets a Boolean value indicating if this DdsRecord control is a window. The default is **False**.

```
BegProp Window Access(*Public) Type(*Boolean)  
    BegGet; BegSet
```

## Property Value

Boolean. **True** indicates the record is a window; otherwise **False**.

## Remarks

If **PopUpWindows** property is **TRUE** and this property is also **True**, this record will render as a pop-up window.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecordClass](#) | [DdsRecord Class Members](#) | [PopUpWindows Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## WindowHeight Property

Returns or sets the height of the window in pixels.

```
BegProp WindowHeight Access(*Public) Type(*Integer)  
    BegGet; BegSet
```

## Property Value

Integer. The height of the window in pixels.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsRecordClass](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## WindowLeft Property

Returns or sets the x-coordinate of the windows left edge in pixels.

```
BegProp WindowLeft Access(*Public) Type(*Integer)  
    BegGet; BegSet
```

## Property Value

Integer. The x-coordinate of the left edge of the window in pixels.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## WindowLeftField Property

Returns or sets the name of the field containing the x-coordinate of the windows left edge in pixels.

```
BegProp WindowLeftField Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

String. The name of the field containing the x-coordinate of the left edge of the window in pixels.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## WindowTitle Property

Returns or sets the text to use as the window title (WDWTITLE).

```
BegProp WindowTitle Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

String. The text for the window title.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## WindowTitleField Property

Returns or sets the name of the field containing the text to use as the window title (WDWTITLE).

```
BegProp WindowTitleField Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

String. The name of the field containing the text to use as the window title.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsRecord Class | DdsRecord Class Members | ASNA.Monarch.WebDspF Namespace

---

## WindowTop Property

Returns or sets the top position of the window in pixels.

```
BegProp WindowTop Access(*Public) Type(*Integer)  
    BegGet; BegSet
```

### Property Value

Integer. The top position of the window in pixels.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsRecord Class | DdsRecord Class Members | ASNA.Monarch.WebDspF Namespace

---

## WindowTopField Property

Returns or sets the name of the field containing the top edge of the window in pixels.

```
BegProp WindowTopField Access(*Public) Type(*String)  
    BegGet; BegSet
```

### Property Value

String. The name of the field containing the top edge of the window in pixels.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsRecord Class | DdsRecord Class Members | ASNA.Monarch.WebDspF Namespace

## WindowWidth Property

Returns or sets the width of the window in pixels.

```
BegProp WindowWidth Access(*Public) Type(*Integer)  
    BegGet; BegSet
```

### Property Value

Integer. The width of the window in pixels.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**See Also**

[DdsRecord Class](#) | [DdsRecord Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## DdsSubfile Class

**DdsSubfile** is a derived class that represents a Subfile control.

For a list of all members of this type, see [DdsSubfile Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.DdsSubfile

```
public class DdsSubfile Inherits DdsRecord
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **DdsSubfile** represents a Subfile control.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ASNA.Monarch.WebDspF Namespace](#) | [DdsSubfile Members](#)

---

## DdsSubfile Constructor()

This method constructs a new instance of a **DdsSubfile** object.

```
BegConstructor DdsSubfile Access(*Public)
```

### Returns

A new instance of the **DdsSubfile** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## CreateChildControls Method

This protected method notifies server controls to create any child controls they contain in preparation for posting back or rendering. Overrides **Control.CreateChildControls**.

```
BegFunc CreateChildControls Modifier(*Overrides) Access(*Protected) Type(Void)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

---

## GetConditionedValue Method

This method returns a string containing the conditional values for a specific field.

```
BegFunc GetConditionedValue Access(*Public) Type(*String)
    DclSrParm fieldID Type(*string)
    DclSrParm possibleValues Type(ASNA.Monarch.WebDspF.ConditionalValue) Rank(1)
```

## Parameters

### fieldID

String containing the ID for the field to get the conditional values for.

### possibleValues

An array of ASNA.**Monarch.WebDspF.ConditionalValue** objects that represents the conditions to return.

## Returns

String containing the conditional indicator expressions for the specific field. The composition of a conditional indicator expression is **value:condition**. An example using conditional indicators might be Red:03 & !99, where **value** is "Red", and the **condition** is "03 & !99".

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfile Class | DdsSubfile Class Members | ConditionalValue Class | ASNA.Monarch.WebDspF Namespace

---

## IsNotFalse Method

This method returns **True** if the condition for the fieldID is not false.

```
BegFunc IsNotFalse Access(*Public) Type(*Boolean)
    DclSrParm fieldID Type(*string)
    DclSrParm condition Type(*string)
```

## Parameters

### fieldID

String containing the ID for the field to evaluate.

### condition

String containing the condition to evaluate.

## Returns

**True** if the *condition* for the *fieldID* is not false; otherwise **False**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [ConditionalValue Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

## IsTrue Method

This method returns **True** if the condition for the fieldID evaluates true.

```
BegFunc IsTrue Access(*Public) Type(*Boolean)
    DclSrParm fieldID Type(*string)
    DclSrParm condition Type(*string)
```

## Parameters

### fieldID

String containing the ID for the field to evaluate.

### condition

String containing the condition to evaluate.

## Returns

**True** if the *condition* for the *fieldID* evaluates true; otherwise **False**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [ConditionalValue Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

## LoadPostData Method

Processes post back data for an ASP.NET server control.

```
BegFunc LoadPostData Access(*Public) Modifier(*Overrides) Type(*Boolean)
    DclSrParm postDataKey Type(*String)
    DclSrParm values Type(System.Collections.Specialized.NameValueCollection)
```

## Parameters

### postDataKey

String containing the identifier for the control.

### values

**System.Collections.Specialized.NameValueCollection** containing a collection of all incoming name values. Represents a sorted collection of associated System.String keys and System.String values that can be accessed either with the key or with the index.

## Returns

Returns **True** if the server control's state changed as a result of the postback; otherwise **False**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## OnInit Method

This protected method raises the **Init** event (in the ASP.NET lifecycle) and returns the **System.EventArgs** object containing the event data. Overrides **Control.OnInit**.

```
BegSr OnInit Modifier(*Overrides) Access(*Protected) Type(Void)
    DeclSrParm e Type (System.EventArgs)
```

## Parameters

**e**

A reference to a **System.EventArgs** object that contains the event data.

## Remarks

This is the initial phase of a server control lifecycle when the server control is initialized. When notified by this method, server controls must perform any initialization steps that are required to create and set up an instance. The view state has not been populated and other server controls are not ready for access.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## RaisePostDataChangedEvent Method

This method signals the server control object to notify the ASP.NET application that the state of the control has changed.

```
BegFunc RaisePostDataChangedEvent Access(*Public) Modifier(*Overrides)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ReadBrowser Method

This method reads the browser.

```
BegSr ReadBrowser Access(*Public) Type(Void)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

---

## RenderChildren Method

This protected method outputs the content of a server control's children to a **System.Web.UI.HtmlTextWriter** object, which writes the content to be rendered on the client. Overrides **Control.RenderChildren**.

```
BegSr RenderChildren Modifier(*Overrides) Access(*Protected) Type(Void)  
    DclSrParm writer Type(System.Web.UI.HtmlTextWriter)
```

## Parameters

**writer**

The **System.Web.UI.HtmlTextWriter** object that receives the rendered content.

## Remarks

This method notifies ASP.NET to render any ASP code on the page. If no ASP code exists, any child controls for the server control are rendered.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

---

## ResetNextChanged Method

This method resets the **NextChanged** property to the default value (\*False).

```
BegFunc ResetNextChanged Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

---

## NextChanged Property

Returns or sets a string containing \*False or \*True; indicating if READC will read this record. (SFLNXTCHG)

```
BegProp NextChanged Access(*Public) Type(*String)  
    BegGet; BegSet
```

## Property Value

String. **\*False** indicates READC will not be used to read this record, **\*True** indicates READC will read this record.

### Remarks

The default is **\*False**, indicating that READC will not be used to read this record. Set to **\*True** to guarantee that a READC will read this record.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

---

## recordCount Property

Returns or sets an integer value containing the number of records in the subfile.

```
BegProp recordCount Access(*Protected) Type(*Integer)
    BegGet; BegSet
```

### Property Value

Integer containing the number of records in the subfile.

### Remarks

The default is **\*False**, indicating that READC will not be used to read this record. Set to **\*True** to guarantee that a READC will read this record.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

---

## subfileStyle Property

Returns or sets the style of the Subfile from a member of the **SubfileStyles** enumeration that specifies the HTML element used for the subfile data; Classic, Checkboxes, DropDown, ListBox, and RadioButtons.

```
BegProp subfileStyle Access(*Protected) Type(SubfileStyles)
    BegGet; BegSet
```

### Property Value

**SubfileStyles** that specifies the HTML element used for the subfile data; Classic, Checkboxes, DropDown, ListBox, and RadioButtons.

### Remarks

The default is **Classic**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**See Also**

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [SubfileStyles Enumeration](#) | [ASNA.Monarch.WebDspF Namespace](#)

This page intentionally left blank.

## DdsSubfileControl Class

The **DdsSubfileControl** class is an extension of a **DdsRecord** for the Subfile. It is both the controller and container of a subfile.

For a list of all members of this type, see [DdsSubfileControl Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.DdsSubfileControl

```
public class DdsSubfileControl Inherits DdsRecord
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

An instance of **DdsSubfileControl** represents an extension of a

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ASNA.Monarch.WebDspF Namespace](#) | [DdsSubfileControl Members](#)

## DdsSubfileControl Constructor()

This method constructs a new instance of a **DdsSubfileControl** object.

```
BegConstructor DdsSubfileControl Access(*Public)
```

## Returns

A new instance of the **DdsSubfileControl** object with default values.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[DdsSubfileControl Class](#) | [DdsSubfileControl Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## OnInit Method

This protected method raises the **Init** event (in the ASP.NET lifecycle) and returns the **System.EventArgs** object containing the event data. Overrides **Control.OnInit**.

```
BegSr OnInit Modifier(*Overrides) Access(*Protected) Type(Void)
DclSrParm e Type (System.EventArgs)
```

## Parameters

**e**

A reference to a **System.EventArgs** object that contains the event data.

### Remarks

This is the initial phase of a server control lifecycle when the server control is initialized. When notified by this method, server controls must perform any initialization steps that are required to create and set up an instance. The view state has not been populated and other server controls are not ready for access.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfileControl Class](#) | [DdsSubfileControl Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### Render Method

This protected method outputs the content of a server control's to a **System.Web.UI.HtmlTextWriter** object, which writes the markup content to be rendered on the client. Overrides **Control.Render**.

```
BegSr Render Modifier(*Overrides) Access(*Protected) Type(Void)
    DeclSrParm writer Type(System.Web.UI.HtmlTextWriter)
```

### Parameters

**writer**

The **System.Web.UI.HtmlTextWriter** object that receives the rendered content.

### Remarks

This method notifies ASP.NET to render any ASP code on the page. If no ASP code exists, any controls for the server control are rendered.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfileControl Class](#) | [DdsSubfileControl Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### ResetSubfileMessage Method

This method resets the **SubfileMessage** property to default values.

```
BegFunc ResetSubfileMessage Access(*Public)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfile Class](#) | [DdsSubfile Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## ResetSubfileMessageId Method

This method resets the **SubfileMessageId** property to default values.

```
BegFunc ResetSubfileMessageId Access(*Public)
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

## scrollDownControls Method

This method sets new scroll updown control values for rendering output.

```
BegSr scrollDownControls Access(*Public)
  DclSrParm edge Type(*integer)
  DclSrParm increment Type(*integer)
```

### Parameters

#### edge

Integer. The value indicating the top edge.

#### increment

Integer. The value that determines the amount by which **Value** changes when the **UpDown** control buttons are clicked.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsSubfile Class | DdsSubfile Class Members | ASNA.Monarch.WebDspF Namespace

## ClearRecords Property

Returns or sets a string value containing an indicator expression that when turned on, clears the data of the detail records (SFLCLR).

```
BegProp ClearRecords Access(*Public) Type(*String)
  BegGet; BegSet
```

### Property Value

String. A string value containing an indicator expression that when turned on, clears the data of the detail records.

### Remarks

The **ClearRecords** property works similar to the SFLCLR keyword on the subfile control record format so that your program can clear the subfile of all records. This property differs from the **InitializeRecords** property (SFLINZ keyword) in that after being cleared, the subfile contains no data. Clearing the subfile does not affect the display. However, after being cleared, the subfile contains no active records.

When active records already exist in the subfile and all are to be replaced, your program can send an output operation to the subfile control record format after selecting SFLCLR. This clears the subfile and permits your

program to write new records to the subfile (by issuing output operations to the subfile record format while incrementing the relative record number). Issuing an output operation to an already active subfile record causes an error message to be returned to your program.

If **ClearRecords** is in effect on an output operation and no records exist in the subfile, the ClearRecords indicator expression is ignored.

An option indicator is required for this Property to prevent the OS/400 program from clearing the subfile on every output operation to the SubfileControl record format.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfileControl Class](#) | [DdsSubfileControl Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ControlledRecord Property

Returns or sets a string value containing the name of the subfile record format that is to be a subfile control record format (SFLCTL).

```
BegProp ControlledRecord Access(*Public) Type(*String)
    BegGet; BegSet
```

### Property Value

String. A string value containing the name of the subfile control record format.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfileControl Class](#) | [DdsSubfileControl Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## DisplayFields Property

Returns or sets a string value containing an indicator expression that when turned on, the fields in this format are displayed to the user (SFLDSPCTL).

```
BegProp DisplayFields Access(*Public) Type(*String)
    BegGet; BegSet
```

### Property Value

String. A string value containing an indicator expression that when turned on, the fields in this format are displayed to the user.

### Remarks

The **DisplayFields** property works similar to the **SFLDSPCTL** keyword so that your program displays fields in the subfile control record format when your program sends an output operation to the subfile control record format. If you do not use an option indicator with this keyword, such as **!90**, the subfile control record is displayed on every output operation to the subfile control record format.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfileControl Class | DdsSubfileControl Class Members | ASNA.Monarch.WebDspF Namespace

---

## DisplayRecords Property

Returns or sets a string value containing an indicator expression, that when turned on, the fields in this format are displayed to the user (SFLDSP).

```
BegProp DisplayRecords Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. A string value containing an indicator expression, such as **!90**, that when turned on, the fields in this format are displayed to the user.

## Remarks

The **DisplayRecords** property works similar to the **SFLDSP** keyword so that the program displays the subfile when your program sends an output operation to the SubfileControl record format.

Use the **SubfilePage** property to specify the number of records in the Subfile to be displayed at the same time.

If your program sends an output operation to the record format when the DisplayRecords indicator is in effect and the subfile is not activated (by adding records to it or by using SFLINZ), an error message is sent to your program.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfileControl Class | DdsSubfileControl Class Members | ASNA.Monarch.WebDspF Namespace

---

## InitializeRecords Property

Returns or sets a string value containing an indicator expression that when turned on (\*True), all of the records in the Subfile are initialized (SFLINZ).

```
BegProp InitializeRecords Access(*Public) Type(*String)
    BegGet; BegSet
```

## Property Value

String. A string value containing an indicator expression that when turned on (\*True), all of the records in the Subfile are initialized.

## Remarks

The **InitializeRecords** property works similar to the **SFLINZ** keyword to initialize all records in the subfile on an output operation to the SubfileControl record format.

When the subfile is displayed (on an output operation to the SubfileControl record), all records in the subfile are displayed with the same value. Any record previously written is overwritten and no longer has its earlier value.

After your program sends an output operation to the SubfileControl record with the InitializeRecords indicator in effect, all records in the subfile are considered active but not changed. They are considered changed only when the indicator specified with the Subfile's **ChangeInd** property.

In general, specify InitializeRecords with the SubfileControl's **ProgramQ** property so that your program can build a message subfile with a single output operation.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfileControl Class](#) | [DdsSubfileControl Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ProgramQLen Property

Returns or sets the length of the **ProgramQ** property field.

```
BegProp ProgramQLen Access(*Public) Type(*Integer)
    BegGet; BegSet
```

### Property Value

Integer. The length of the field specified by the **ProgramQ** property.

### Remarks

The values you can use for ProgramQLen are:

- **10 (Default)** - Generates a 10-byte character field - \*Char(10).
- **276** - Generates a 276-byte character field - \*Char(276).
- **0** - Use for String fields - \*String.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[DdsSubfileControl Class Members](#) | [ProgramQ Property](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ProgramQ Property

Returns or sets the name of the field containing the name of the program queue used to build a message subfile (SFLPGMQ).

```
BegProp ProgramQ Access(*Public) Type(*String)
    BegGet; BegSet
```

### Property Value

String. The name of the field containing the name of the program queue used to build a message subfile.

### Remarks

The **ProgramQ** property works similar to the **SFLPGMQ** keyword to specify the name of the field containing the name of the program queue used to build a message subfile.

An entry in this property of either the field name of the program queue, or the special value **\*PGM** indicates that the Subfile is now a MessageSubfile.

In the **ProgramQ** property, specify either the name of the field that contains the message subfile queue, or the special value **\*PGM**, which specifies to use the message queue of the program issuing the output operation.

The first field in the subfile record format is used for the Message subfile. This field contains the name of the program message queue used by the program to build a message subfile. In addition, ProgramQ can be specified on the SubfileControl record format when the SubfileControl's **InitializeRecords** property is enabled.

The length of this field is specified by the **ProgramQLen** property, which has a default length of 10 generating a 10-byte character field. See **SubfileControl.ProgramQLen** property for more information.

When you specify a field name in the **ProgramQ** property, and also enable the **InitializeRecords** property, the Message Subfile is initialized with all the messages that are in the program message queue for the name or current program (**\*PGM**) specified in the **ProgramQ** property.

The **ProgramQ** field can also contain the special value **\*PGM** instead of a program message queue name. **\*PGM** indicates to use the message queue of the program issuing the output operation.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfileControl Class | DdsSubfileControl Class Members | SubfileControl.InitializeRecords Property | SubfileControl.ProgramQLen Property | ASNA.Monarch.WebDspF Namespace

## SubfileMessageId Property

Returns or sets a new instance of an **ErrMsgIdProperty** object that identifies a message ID number to be associated with this field.

```
BegProp SubfileMessageId Access(*Public) Type(ErrMsgIdProperty)
    BegGet; BegSet
```

## Property Value

**ErrMsgIdProperty** object containing the conditional property values for the field as shown below

## Remarks

The **SubfileMessageId** property displays the SubfileMessageId dialog, by selecting the ... button to the right; as shown below.



## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfileControl Class | DdsSubfileControl Class Members | ASNA.Monarch.WebDspF Namespace

## SubfileMessage Property

Returns or sets a new instance of an **ErrMsgProperty** object that identifies a message to be associated with this field.

```
BegProp SubfileMessage Access(*Public) Type(ErrMsgProperty)
BegGet; BegSet
```

## Property Value

ASNA.Monarch.WebDspF.ErrorMsgProperty. Returns or sets an instance of the ASNA.Monarch.WebDspF.SubfileMessageProperty class, which displays the **SubfileMessage Property Editor Dialog**, as shown below.

## Remarks

The **SubfileMessage** property displays the SubfileMessage dialog, by selecting the ... button to the right; as shown below.



## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

DdsSubfileControl Class | DdsSubfileControl Class Members | ASNA.Monarch.WebDspF Namespace

## SubfilePage Property

Returns or sets the number of records in the Subfile to be displayed at the same time (SFLPAG).

```
BegProp SubfilePage Access(*Public) Type(*Integer)
    BegGet; BegSet
```

### Property Value

Integer specifying the number of records in the Subfile to be displayed at the same time.

### Remarks

The **SubfilePage** property works similar to the **SubfilePage** keyword that specifies the number of records in the Subfile to be displayed at the same time.

The default value is **0**.

The **SubfilePage** value and the number of lines required by each Subfile record determine the number of actual lines required to display the page of records. Not all records within a Subfile must be displayed at the same time, and not all lines of the display are required to display a page of Subfile records.

When you specify the same values for the **SubfilePage** and **SubfileSize** properties, the maximum number of records that can be contained in the Subfile equals the maximum number of Subfile records that can appear on the display at one time. You can also specify option indicators for fields in the Subfile record format.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsSubfileControl Class | DdsSubfileControl Class Members | SubfileControl.SubfileSize Property | ASNA.Monarch.WebDspF Namespace

## SubfileSize Property

Returns or sets the number of records in the Subfile (SFLSIZ).

```
BegProp SubfileSize Access(*Public) Type(*Integer)
    BegGet; BegSet
```

### Property Value

Integer. Specifies the number of records in the Subfile.

### Remarks

The **SubfileSize** property works similar to the **SubfileSize** keyword to specify the number of records in the Subfile.

The default value is **0**. The maximum number of records allowed is 9999.

SubfileSize equals SubfilePage:

If you specify the same values for the **SubfileSize** and **SubfilePage** properties, you can specify option indicators for fields in the Subfile record format. (This is called field selection.)

When the Subfile is built, the records can vary in length depending upon which fields are selected, and each output operation places records into successive positions within the Subfile. Each record in the Subfile can require a different number of display lines. The number of records that actually fit into the Subfile depends upon the fields selected for each record written to the Subfile.

The **SubfilePage** property value is increased to equal the maximum number of records that fit on the display if the number of Subfile records to be displayed do not occupy a full display.

SubfileSize does not equal SubfilePage:

When you specify different values for the **SubfilePage** and **SubfileSize** properties, the **SubfileSize** value specifies the number of records that can be placed into the Subfile. If your program places a record with a relative record number larger than the **SubfileSize** value into the Subfile, the Subfile is automatically extended to contain it (up to a maximum of 9999 records). The value you specify should be large enough to accommodate the maximum number of records you would normally have in the Subfile.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsSubfileControl Class | DdsSubfileControl Class Members | SubfileControl.SubfilePage Property | ASNA.Monarch.WebDspF Namespace

---

## SubfileStyle Property

Returns or sets the style of the Subfile from a member of the **SubfileStyles** enumeration that specifies the HTML element used for the subfile data; Classic, Checkboxes, DropDown, ListBox, and RadioButtons.

```
BegProp SubfileStyle Access(*Public) Type(SubfileStyles)  
BegGet; BegSet
```

### Property Value

**SubfileStyles** that specifies the HTML element used for the subfile data; Classic, Checkboxes, DropDown, ListBox, and RadioButtons.

### Remarks

The default subfile style is **Classic**, which functions the same as subfiles in AVR Classic. Click on the arrow to the right, or double-click the field to select or display the other options of **Checkboxes**, **DropDown**, **ListBox**, and **RadioButtons**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

DdsSubfileControlClass | DdsSubfileControl Class Members | SubfileControl.InitializeRecords Property | SubfileControl.SubfileStyleLen Property | ASNA.Monarch.WebDspF Namespace

---

## EraseProperty Class

The **EraseProperty** is a derived class that is a container for the EraseFormat Property Editor Dialog values. For a list of all members of this type, see EraseProperty Members.

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.EraseProperty

```
Public class EraseProperty Inherits ConditionalProperty
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **EraseProperty** represents an array of record formats/conditions you want to be erased from the display when this record is written. The default condition is **\*ALL**, which indicates to remove all records.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace | EraseProperty Members

---

## EraseProperty Constructor()

This method constructs a new instance of an **EraseProperty** object.

```
BegConstructor EraseProperty Access(*Public)
```

### Returns

A new instance of the **EraseProperty** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

EraseProperty Class | EraseProperty Class Members | ASNA.Monarch.WebDspF Namespace

---

## EraseProperty Constructor(String)

This method constructs a new instance of an **EraseProperty** object with values established.

```
BegConstructor EraseProperty Access(*Public)  
  DclSrParm propString Type(*String)
```

### Parameters

propString

String containing an array of record format / conditions you want to be erased from the display when this record is written.

### Returns

A new instance of the **EraseProperty** object with values established.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[EraseProperty Class](#) | [EraseProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### GetValidValues Method

Returns a string array of the values that are valid for the **EraseProperty** object .

```
BegFunc GetValidValues Access(*Public) Type(*String) Rank(1) Modifier(*overrides)
```

### Returns

A string array of the values that are valid.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[EraseProperty Class](#) | [EraseProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### NewInstance Method

Creates a new instance of a **ConditionalProperty** object modeling the value/conditions represented by

```
BegFunc NewInstance Access(*Public) Type(ConditionalProperty) Modifier(*overrides)
```

### Returns

A new instance of a **ConditionalProperty**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[EraseProperty Class](#) | [EraseProperty Class Members](#) | [ConditionalProperty Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ErrMsgIdProperty Class

The **ErrMsgIdProperty** is a derived class that is a container for the Error Message ID Property Editor Dialog values.

For a list of all members of this type, see [ErrMsgIdProperty Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.ErrMsgIdProperty

```
Public class ErrMsgIdProperty Inherits ConditionalProperty
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **ErrMsgIdProperty** represents an array of value/conditions error messages using a message file.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ASNA.Monarch.WebDspF Namespace](#) | [ErrMsgIdProperty Members](#)

---

## ErrMsgIdProperty Constructor()

This method constructs a new instance of an **ErrMsgIdProperty** object.

```
BegConstructor ErrMsgIdProperty Access(*Public)
```

### Returns

A new instance of the **ErrMsgIdProperty** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ErrMsgIdProperty Class](#) | [ErrMsgIdProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## ErrMsgIdProperty Constructor(String)

This method constructs a new instance of an **ErrMsgIdProperty** object with values established.

```
BegConstructor ErrMsgIdProperty Access(*Public)  
DclSrParm propString Type(*String)
```

### Parameters

#### propString

String containing an array of value/conditions error messages using a message file.

### Returns

A new instance of the **ErrMsgIdProperty** object with values established.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ErrMsgIdProperty Class](#) | [ErrMsgIdProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### GetValidValues Method

Returns a string array of the values that are valid for the **ErrMsgIdProperty** object .

```
BegFunc GetValidValues Access(*Public) Type(*String) Rank(1) Modifier(*overrides)
```

### Returns

A string array of the values that are valid.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ErrMsgIdProperty Class](#) | [ErrMsgIdProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### NewInstance Method

Creates a new instance of a **ConditionalProperty** object modeling the value/conditions represented by

```
BegFunc NewInstance Access(*Public) Type(ConditionalProperty) Modifier(*overrides)
```

### Returns

A new instance of a **ConditionalProperty**.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ErrMsgIdProperty Class](#) | [ErrMsgIdProperty Class Members](#) | [ConditionalProperty Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

## ErrMsgProperty Class

The **ErrMsgProperty** is a derived class that is a container for the Error Message Property Editor Dialog values. For a list of all members of this type, see [ErrMsgProperty Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.ErrMsgProperty

```
Public class ErrMsgProperty Inherits ConditionalProperty
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **ErrMsgProperty** represents an array of value/conditions error messages for a field control.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ASNA.Monarch.WebDspF Namespace](#) | [ErrMsgProperty Members](#)

## ErrMsgProperty Constructor()

This method constructs a new instance of an **ErrMsgProperty** object.

```
BegConstructor ErrMsgProperty Access(*Public)
```

### Returns

A new instance of the **ErrMsgProperty** object with default values.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[ErrMsgProperty Class](#) | [ErrMsgProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## ErrMsgProperty Constructor(String)

This method constructs a new instance of an **ErrMsgProperty** object with values established.

```
BegConstructor ErrMsgProperty Access(*Public)
  DclSrParm propString Type(*String)
```

### Parameters

#### propString

String containing an array of value/conditions error messages for a field control.

### Returns

A new instance of the **ErrMsgProperty** object with values established.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ErrMsgProperty Class](#) | [ErrMsgProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## GetValidValues Method

Returns a string array of the values that are valid in

```
BegFunc GetValidValues Access(*Public) Type(*String) Rank(1) Modifier(*overrides)
```

## Returns

A string array of the values that are valid.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ErrMsgProperty Class](#) | [ErrMsgProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## NewInstance Method

Creates a new instance of a **ConditionalProperty** object modeling the value/conditions represented by representing the

```
BegFunc NewInstance Access(*Public) Type(ConditionalProperty) Modifier(*overrides)
```

## Returns

A new instance of a **ConditionalProperty** object represented by the **ErrMsgProperty** object.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ErrMsgProperty Class](#) | [ErrMsgProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## KeyProperty Class

The **KeyProperty** is a derived class that is a container for the EraseFormat Property Editor Dialog values.

For a list of all members of this type, see [KeyProperty Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF.KeyProperty

```
Public class KeyProperty Inherits ConditionalProperty
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

An instance of **KeyProperty** represents an array of record formats/conditions you want to be erased from the display when this record is written. The default condition is **\*ALL**, which indicates to remove all records.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ASNA.Monarch.WebDspF Namespace](#) | [KeyProperty Members](#)

---

## KeyProperty Constructor()

This method constructs a new instance of a **KeyProperty** object.

```
BegConstructor KeyProperty Access(*Public)
```

## Returns

A new instance of the **KeyProperty** object with default values.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[KeyProperty Class](#) | [KeyProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## KeyProperty Constructor(String)

This method constructs a new instance of a **KeyProperty** object with indicator expression values established.

```
BegConstructor KeyProperty Access(*Public)  
DclSrParm indicatorExpressionString Type(*String)
```

## Parameters

### indicatorExpressionString

String containing an array of record format / conditions you want to be erased from the display when this record is written.

## Returns

A new instance of the **KeyProperty** object with values established.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[KeyProperty Class](#) | [KeyProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## KeyProperty Constructor(String)

This method constructs a new instance of an **KeyProperty** object with values established.

```
BegConstructor KeyProperty Access(*Public)  
  DeclSrParm propString Type(*String)
```

## Parameters

### propString

String containing an array of record format / conditions you want to be erased from the display when this record is written.

## Returns

A new instance of the **KeyProperty** object with values established.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[KeyProperty Class](#) | [KeyProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## GetValidValues Method

Returns a string array of the values that are valid for the **KeyProperty** object .

```
BegFunc GetValidValues Access(*Public) Type(*String) Rank(1) Modifier(*overrides)
```

## Returns

A string array of the values that are valid.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[KeyProperty Class](#) | [KeyProperty Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## NewInstance Method

Creates a new instance of a **ConditionalProperty** object modeling the value/conditions represented by

```
BegFunc NewInstance Access(*Public) Type(ConditionalProperty) Modifier(*overrides)
```

## Returns

A new instance of a **ConditionalProperty**.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[KeyProperty Class](#) | [KeyProperty Class Members](#) | [ConditionalProperty Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

This page intentionally left blank.

---

## Page Class

Monarch display files are based on the ASP.Net **System.Web.UI.Page**. The Monarch framework defines the class **Page** to enhance the behavior of the system's page by adding certain properties and events to the base class.

For a list of all members of this type, see [Page Members](#).

ASNA.Monarch.WebDspF

### ASNA.Monarch.WebDspF. Page

```
BegClass Page Access(*public) Inherits(System.Web.UI.Page)
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

Besides the normal phases that an ASPX goes through in ASP.NET, there are enhanced or additional phases specific to Monarch that a display file goes through. These phases take place when the page is in the ASP.NET Prerender (see [Control Execution Lifecycle](#)) phase.

Use the **OnCopyDspFileToBrowser** method to intervene when data is being copied from the display buffer data sent to the 'dds' controls in the page (Response is formulated here).

Use the **OnCopyBrowserToDspFile** method to intervene when data is being copied from the data sent by the Browser to the display buffer data set. (Request is processed here).

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ASNA.Monarch.WebDspF Namespace](#) | [Page Members](#)

---

## Page Constructor ()

Creates a new instance of a **Page** object setting any server control fields to their respective default values.

```
BegFunc Page() Access(*Public)
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## GetFileName Method

Gets the name of the file on the page.

```
BegFunc GetFileName Modifier(*Overrides) Access(*Protected)
```

## Returns

**String.** The name of the file on the page.

## Exceptions

None.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## OnCopyBrowserToDspFile Method

Allows you to intervene when data is being copied from the data sent by the browser to the display buffer data set. The **Request** is processed here.

```
BegFunc OnCopyBrowserToDspFile Modifier(*Overrides) Access(*Protected)
```

## Exceptions

None.

## Remarks

The **OnCopyBrowserToDspFile** method formulates a request from the Web Browser to the Web Server.

## Examples

```
BegSr OnCopyDspFileToBrowser Modifier(*Overrides) Access(*Protected)
  If (*Not IsPostBack)

    // Specify the data source and field names for the Text
    // and Value properties of the items (ListItem objects)
    // in the DropDownList control.
    CardexList.DataSource      = CreateDataSource()
    CardexList.DataTextField  = "CardexTextField"
    CardexList.DataValueField = "CardexValueField"

    // Bind the data to the control.
    CardexList.DataBind()

    // Set the default selected item, if desired.
    CardexList.SelectedIndex = 0

  EndIf
  *base.OnCopyDspFileToBrowser()
EndSr
```

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## OnCopyDspFileToBrowser Method

Allows you to intervene when data is being copied from the display buffer data sent to the 'dds' controls in the page. The Response is formulated here.

```
BegFunc OnCopyDspFileToBrowser Modifier(*Overrides) Access(*Protected)
```

### Exceptions

None.

### Remarks

The **OnCopyDspFileToBrowser** method formulates a response from the Web server to the Web Browser.

### Examples

```
BegSr OnCopyDspFileToBrowser Modifier(*Overrides) Access(*Protected)
  If (*Not IsPostBack)

    // Specify the data source and field names for the Text
    // and Value properties of the items (ListItem objects)
    // in the DropDownList control.
    CardexList.DataSource      = CreateDataSource()
    CardexList.DataTextField  = "CardexTextField"
    CardexList.DataValueField = "CardexValueField"

    // Bind the data to the control.
    CardexList.DataBind()

    // Set the default selected item, if desired.
    CardexList.SelectedIndex = 0

  EndIf
  *base.OnCopyDspFileToBrowser()
EndSr
```

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

## OnInit Method

Overrides **Control.OnInit**. Raises the **Init** event (in the ASP.NET lifecycle) and returns the **System.EventArgs** object containing the event data.

```
BegSr OnInit Modifier(*Overrides) Access(*Protected) Type(Void)
  DeclSrParm e Type (System.EventArgs)
```

### Parameters

e

A reference to a **System.EventArgs** object that contains the event data.

## Remarks

This is the initial phase of a server control lifecycle when the server control is initialized. When notified by this method, server controls must perform any initialization steps that are required to create and set up an instance. The view state has not been populated and other server controls are not ready for access.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## SaveViewState Method

Overrides **Control.SaveViewState**. Saves the web control view-state changes made since the time the Page was posted back to the server.

```
BegFunc SaveViewState() Modifier(*Overrides) Access(*Protected) Type(Object)
```

## Return Value

Returns the object that contains the web control's current view state. If there is no view state associated with the control, this method returns a null reference.

## Exceptions

None.

## Remarks

When view state is saved, this string object is returned to the client as a variable in an HTML "Hidden" element.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## DataSet Property

Returns an instance of the **System.Data.DataSet** class which represents an in-memory cache of data.

```
BegProp DataSet Access(*Public) Type(System.Data.DataSet)  
BegGet
```

## Returns

**System.Data.DataSet** . Represents an in-memory cache of data.

## Remarks

A DataSet is a buffer containing one DataTable per record format in the File. A DataTable contains one record per regular format.

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## OutOfSequence Property

Returns or sets a Boolean value indicating the request from the browser is out of sequence.

```
BegProp OutOfSequence Access(*Public) Type(*Boolean)
BegGet; BegSet
```

## Property Values

Boolean. **True** if the request from the browser is out of sequence; otherwise **False**.

## Remarks

This property gets established in the page Load phase. The typical cause of out of sequence conditions is if the user hits the back button or jumped to a previously bookmarked page. A **True** setting can prevent the copy-browser-to-display-file and the process-business-rules phases from executing (ControlExecutionLifeCycle).

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## RunProgram Property

Returns or sets a Boolean value indicating whether to pass control to the AVR program waiting on an **EXFMT**.

```
BegProp RunProgram Access(*Public) Type(*Boolean)
BegGet; BegSet
```

## Property Values

Boolean. **True** to pass control to the AVR program waiting on an **EXFMT**; otherwise **False**.

## Remarks

This property gets set **True** at page Load if the request is in sequence (**OutOfSequence** False) and an attention key was pressed. This property can be set **False** in any phase after page Load and before process-business-rules (ControlExecutionLifeCycle).

## Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### Device Field

Returns an instance of an **ASNA.MonarchWebDevice** object which represents the web device.

```
BegFunc Device Access(*Protected) Type(ASNA.Monarch.WebDevice)
```

### Returns

**ASNA.Monarch.WebDevice**. Represents the web device object.

### Exceptions

None.

### Requirements

Namespace: ASNA.Monarch.WebDspF

Assembly: ASNA.Monarch.WebDspF.dll

Platforms: Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDevice Class](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

### DspF Field

Returns an instance of an **ASNA.Monarch.WebDisplayFile** object which represents the display file.

```
BegFunc DspF Access(*Protected) Type(ASNA.Monarch.WebDisplayFile)
```

### Returns

**ASNA.Monarch.WebDisplayFile**. Represents the display file object.

### Exceptions

None.

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Page Class](#) | [Page Class Members](#) | [ASNA.Monarch.WebDspF Namespace](#)

---

## Enumerations

---

### BannerStyles Enumeration

The **BannerStyles** enumerated constant defines how to display function keys on the page.

```
BegEnum BannerStyles Access(*Public)
```

#### Members

Member	Description	Value
FullBanner	The display is a full display.	0
Horizontal	The display is horizontal. This is the default.	1
Vertical	The display is vertical.	2

#### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

#### See Also

ASNA.Monarch.WebDspF Namespace

## DateTimeFormat Enumeration

The **DateTimeFormat** enumerated constant defines values describing the format of a date timestamp field.

```
BegEnum DateTimeFormat Access(*Public)
```

### Remarks

DateTimeFormat defines values in which you can select one of the choices.

### Members

Member	Description	Value
DMY	Day/Month/Year. For date, dd/mm/yy.	8
EUR	IBM European Standard. For date, dd.mm.yyyy. For time, hh:mm:ss.	4
HMS	Hours/Minutes/Seconds. For time, hh:mm:ss.	10
ISO	International Standards Organization. For date, yyyy-mm-dd. For time, hh:mm:ss.	2
JIS	Japanese Industrial Standard Christian Era. For date, yyyy-mm-dd. For time, hh:mm:ss.	5
JUL	Julian. For date, yy/ddd.	9
MDY	Month/Day/Year. For date, mm/dd/yy.	7
USA	IBM USA Standard. For date, mm/dd/yyyy. For time, hh:mm AM.	3
YMD	Year/Month/Day. For date, yy/mm/dd.	6

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**Assembly:** ASNA DataGate Client (in ASNA.DataGate.Client.dll)

### See Also

ASNA.Monarch.WebDspF Namespace

## DecType Enumeration

The **DecType** enumerated constant defines the type of decimal field. The default is **Zoned**.

```
BegEnum DecType Access(*Public)
```

### Members

Member	Description	Value
Binary	The field is binary.	0
Packed	The field is packed.	1
Zoned	The field is zoned. This is the default.	2

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

## FieldUsages Enumeration

The **FieldUsages** enumerated constant defines how the field is used.

```
BegEnum FieldUsages Access(*Public)
```

### Members

Member	Description	Value
Hidden	The field is hidden. These fields are passed from and to the program but are not sent to the display. Hidden fields are useful in applications involving subfiles. For example, a subfile record can contain record key information in a hidden field. The hidden field cannot be seen by the user, but is returned to program with the subfile record so that the program can return the record to the database.	0
OutputOnly	The field in output only. These fields that are passed from the program to the display station when the program writes a record to a display. Output fields contain data provided by the program, not by the user. In the case of subfiles, which are special records used to display lists of information, output fields are returned to the program as if they were output/input fields.	1
InputOnly	The field is input-only. These fields that are passed from the display station to the program when the program reads a record. They can be initialized with a default value (specified in the record format for the display file). If the user does not change the field and the field is selected for input, the default value is passed to the program. Input fields that are not initialized are displayed as blanks into which the user can enter data. <b>Note:</b> Trailing blanks on input fields are replaced by null and not blank characters; therefore, the Insert key can be used to insert characters in all input fields that end in blanks.	2
Both	The field is output and input. This is the default.	3

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

## InputStyles Enumeration

The **InputStyles** enumerated constant defines the input style for a character field. The default is **Textbox**.

```
BegEnum InputStyles Access(*Public)
```

### Members

Member	Description	Value
Textbox	The field is a single line textbox.	0
MultiLine	The field is a multiple line textbox.	1
Password	The field is a password textbox.	2
Checkbox	The field is checkbox that allows the field to be "checked" or "unchecked".	3

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

## SubfileStyles Enumeration

The **SubfileStyles** enumerated constant defines the "INPUT type=" HTML element used for subfile data.

```
BegEnum SubfileStyles Access(*Public)
```

### Members

Member	Description	Value
Classic	The element is classic style. This is the default.	0
Checkboxes	The element input type is checkbox.	1
Dropdown	The element input type is dropdown.	2
ListBox	The element input type is ListBox.	3
RadioButtons	The element input type is radio.	4

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

## ValuesStyle Enumeration

The **ValuesStyle** enumerated constant defines the control style used to enable users to select from a single selection down-down List box, or Textbox. This impacts the drop down style for the **Values**, **ValuesText**, and the **Text** properties.

```
BegEnum ValuesStyle Access(*Public)
```

### Remarks

A drop-down control shows a drop-down button displaying the associated items, depending upon the **ValuesStyle** type and entries in **Values** and/or **ValuesText**.

### Members

Member	Description	Value
Textbox	(Default). The field displays as a text box.	0
DropDownBoth	The field displays as a drop-down box containing the entries in the <a href="#">Values</a> and <a href="#">ValuesText</a> properties.	1
DropDownText	The field displays as a drop-down box containing the entries in the <a href="#">ValuesText</a> property..	2
DropDownValues	The field displays as a drop-down box containing the entries in the <a href="#">Values</a> property.	3

### Requirements

**Namespace:** ASNA.Monarch.WebDspF

**Assembly:** ASNA.Monarch.WebDspF

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch.WebDspF Namespace

This page intentional left blank.

## ASNA.VisualRPG.Runtime Assembly

### CLProgram Class

The **CLProgram** class provides support within existing CL program procedures in the areas of:

- Library List and File Manipulation
- CL Processing Commands for the Retrieval and Manipulation of Strings and Variables.

For a list of all members of this type, see [CLProgram Members](#).

ASNA.Monarch

ASNA.Monarch.Program

**ASNA.Monarch.CLProgram**

```
BegClass CLProgram Access(public) Inherits(ASNA.Monarch.Program)
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

A CLProgram class contains the following:

#### Library List and File Manipulation

##### Library List

A library list is an ordered set of directory names associated with each applications database connection. The `AddLiblEntry` method adds a library name to the user portion of the library list. The `RmvLiblEntry` method removes a library name from the user portion of the library list.

##### File Overrides

You can use an override command to replace the database file named in a CL procedure or program or to change certain parameters of the existing database file. These overrides can be applied to DBFile, PrintFile, or a WorkStnFile object. This may be especially useful for files that have been renamed or moved since the procedure or program was created. It can also be used to access a file member other than the first member. When you override to a different file(`ToFile`), the overriding file must have only one record format. A logical file which has multiple record formats defined in DDS may be used if it is defined over only one physical file member. A logical file which has only one record format defined in the DDS may be defined over more than one physical file member. The name of the format does not have to be the same as the format name referred to when the program was created. You should ensure that the format of the data in the overriding file is the same as in the original file, otherwise you may get unexpected results. `OverrideFile` is an overloaded method that provides these file overrides with several parameters options. `DeleteOverride` is an overloaded method to remove database file overrides previously applied.

##### Physical File Members

The `ClearPFM` method removes all data from a member of a physical file. The `InitializePFM` overloaded method initializes records in a member of a physical file to the specified type of record (either default or deleted records). If the initialized member is empty, records are added and initialized to the specified type; if the member is not empty, records of the specified type are added to the member. As many records are added as is necessary to make the total record count specified. This method can be used to initialize files that are processed in arrival sequence or by relative record numbers.

#### CL Processing Commands for the Retrieval and Manipulation of Strings and Variables

##### LDA

A local data area is created for each job in the system. The system creates a local data area, which is initially filled with blanks, with a length of 1024 and type \*CHAR. When you submit a job, the value of the submitting job's local data area is copied into the submitted job's local data area. You can use this local data area to pass information to a procedure or program without the use of a parameter list. You can use the overloaded `ChangeDataArea` method to change the value of the current Lda associated with the program, or the overloaded `RetrieveDataArea` to retrieve all or a part of the Lda.

### Job Switches and Attributes

Associated with each job on the iSeries there is a set of properties or attributes. These attributes can be programmatically accessed in CL via the RTVJOB commands. One of these attributes is a series of 8 switches which can be 'on' or 'off'. RPG programs also have access to these switches via the \*INUX indicators. The CLProgram class provides the method RTVJOB\_SWS to facilitate the translation of the RTVJOB CL command SWS attribute. The Percent\_Switch method can be used to compare a bit-mask against the current job switches for equality.

Additionally, there are several methods that can be used to retrieve the job attributes for the job in which a CLProgram is used. These are: RTVJOB\_DATE, RTVJOB\_JOB, RTVJOB\_NBR, RTVJOB\_TYPE, and RTVJOB\_USER and are provided to retrieve the current job date, job name, job number, job type, and current user attributes respectively.

### String Manipulation

The Asterisk\_BCat method concatenates a prefix, variable, and suffix, with a blank between each parameter. TheAsterisk\_TCat method concatenates a prefix, variable, and suffix, with the variable trimmed of leading and trailing blanks. The Percent\_SST method returns a string containing a substring from the contents of the specified CL character variable or the local data area. This can only be used within a CL procedure.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA.VisualRPG.Runtime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch Namespace | CLProgram Class Members

---

## CLProgram Constructor()

Initialize a new instance of a **CLProgram** object.

```
BegFunc CLProgram Access(*Public) Type(CLProgram)
```

### Returns

A new instance of a **CLProgram** object.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class | CLProgram Class Members | ASNA.Monarch Namespace

---

## AddLibEntry Method(string)

This method adds a library name to the user portion of the library list.

```
BegFunc AddLibEntry Access(Protected) Type(CLProgram)  
  Dc1SrParm libraryName Type(*String) Len(45)
```

### Parameters

#### libraryName

String. Specifies the name of the library being added to the user portion of the library list.

## Remarks

This method adds the specified library name to the user portion of the library list (after the current library list entry if it exists) for the process in which the command was entered. Up to 250 libraries may exist in the user portion of the library list. Only one library name is added at a time with this method.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

CLProgram Class: Members, AddLibEntry | ASNA.Monarch Namespace

---

## AddLibEntry Method(string, ASNA.DataGate.Client.LiblPosition, string)

This method add a library name in the to the user portion of the library list as specified by the *position* and *referenceLibraryName* for the process in which the command was entered.

```
BegFunc AddLibEntry Access(Protected) Type(CLProgram)
    DclSrParm libraryName Type(*String)
    DclSrParm position Type(ASNA.DataGate.Client.LiblPosition)
    DclSrParm referenceLibraryName Type(*String)
```

## Parameters

### libraryName

String. Specifies the name of the library being added to the user portion of the library list.

### position

ASNA.DataGate.Client.LiblPosition specifying the position in the user portion of the library list where this library is to be added, relative to the position of the *referenceLibraryName*. If *LiblPosition* is *First*, or *Last*, then the *referenceLibraryName* is ignored. If *LiblPosition* is *Before*, *After*, or *Replace*, then *referenceLibraryName* is used for reference as noted below.

### referenceLibraryName

String. The name of a library that already exists in the user portion of the library list. The library specified in the *libraryName* parameter is added *After* this library, *Before* this library, or *Replaces* this library in the user portion of the library list. If *LiblPosition* is *First*, or *Last*, then this parameter is ignored.

## Remarks

This method adds the specified library name to the user portion of the library list as specified by the *position* and *referenceLibraryName* for the process in which the command was entered. Up to 250 libraries may exist in the user portion of the library list. Only one library name is added at a time with this method. The user can specify whether the library is added to the beginning or the end of the library list. In addition to this, the user can specify whether the library is added before, after, or replaces an existing library in the library list.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

CLProgram Class: Members, AddLibEntry | ASNA.Monarch Namespace

## Asterisk\_BCat Method

Returns a concatenated string.

```
BegFunc Asterisk_BCat Access(Protected) Type(CLProgram)
  DclSrParm prefix Type(*String) Len(45)
  DclSrParm suffix Type(*String) Len(45)
```

### Parameters

**prefix**

String. The string which is to be prefixed to the variable.

**suffix**

String. The string which is to be appended to the variable.

### Returns

String. Concatenated *prefix*, "variable", *suffix* with a blank between each parameter.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class](#) | [CLProgram Class Members](#) | [ASNA.Monarch Namespace](#)

---

## Asterisk\_TCat Method

Returns a concatenated string.

```
BegFunc Asterisk_TCat Access(Protected) Type(CLProgram)
  DclSrParm prefix Type(*String) Len(45)
  DclSrParm suffix Type(*String) Len(45)
```

### Parameters

**prefix**

String. The string which is to be prefixed to the variable.

**suffix**

String. The string which is to be appended to the variable.

### Returns

String. Concatenated *prefix*, "variable", *suffix* with the "variable" trimmed of leading and trailing spaces.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class](#) | [CLProgram Class Members](#) | [ASNA.Monarch Namespace](#)

## ChangeDataArea Method(string, integer, integer, string)

Changes the values of the current data area associated with the program.

```
BegFunc ChangeDataArea Access(Protected) Type(CLProgram)
  DclSrParm dataArea Type(*String) Len(1024)
  DclSrParm start Type(*Integer) Len(4)
  DclSrParm length Type(*Integer) Len(4)
  DclSrParm newValue Type(*String) Len(1024)
```

### Parameters

#### dataArea

String. The data area name.

#### start

Integer. The starting point within the *dataArea*.

#### length

Integer. The length of the data area to be changed.

#### newValue

String. The new data area values for the substitution variables in the *dataArea*. The format of each variable must be defined in the CL program parameters.

### Returns

String. A new instance of the data area with some values, as specified by the *start* position and *length*, replaced with *newValue*.

### Remarks

This method changes values within the *dataArea* but does not change the other attributes.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class | CLProgram Class Members | ASNA.Monarch Namespace

## ChangeDataArea Method(string, string)

Changes the values of the current data area to be associated with the program.

```
BegFunc ChangeDataArea Access(Protected) Type(void)
  DclSrParm dataArea Type(*String) Len(1024)
  DclSrParm newValue Type(*String) Len(1024)
```

### Parameters

#### dataArea

String. The data area name.

#### newValue

String. The new data area values for the substitution variables in the *dataArea*. The format of each variable must be defined in the CL program parameters.

### Returns

String. A new instance of the data area initialized with *newValue*.

## Remarks

This method changes the values of the *dataArea* but does not change the other attributes.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[CLProgram Class](#) | [CLProgram Class Members](#) | [ASNA.Monarch Namespace](#)

---

## ClearPFM Method

This method removes all data from the specified member of a physical file.

```
BegFunc ClearPFM Access(Protected) Type(System.Void)
  DclSrParm File Type(*String) Len(45)
  DclSrParm Mbr Type(*String) Len(45)
```

## Parameters

### File

String. The qualified name of the physical file containing the member to clear. The name of the physical file can be qualified by one of the following library values:

- \*LIBL: All libraries in the job's library list are searched until the first match is found. This is the default.
- \*CURLIB: The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.
- library-name: Specify the name of the library to be searched.
- physical-file-name: Specify the name of the physical file.

### Mbr

String. The optional name of the physical file member within the File to clear. The name of the member can be:

- \*FIRST : The first member in the database file is used. This is the default.
- \*LAST: The last member of the database file is used.
- physical-file-member: Specify the name of the physical file member.

## Remarks

The following restrictions apply:

- The user must have object operational, object management or alter, and delete authority for the physical file that contains the member and execute authority to the library.
- If any of the access paths to the member are in use when this command is entered, or if the physical file member is in use, the command is not run.
- An \*EXCL lock is required on the member to clear it.
- In multithreaded jobs, this command is not thread safe for distributed files. This command is also not thread safe and fails for Distributed Data Management (DDM) files of type \*SNA.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

---

## See Also

CLProgram Class | CLProgram Class Members | InitializePFM Methods | ASNA.Monarch Namespace

---

## DeleteOverride Method(string)

This method removes the database file override previously applied to the

```
BegFunc DeleteOverride Access(*Public) Type(CLProgram)
  DclSrParm fileName Type(*String) Len(45)
```

### Parameters

**fileName**

String containing the database file name to remove overrides from.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

CLProgram Class: Members, OverrideFile | ASNA.Monarch Namespace

---

## DeleteOverride Method(string, ASNA.Monarch.OverrideScope)

This method removes the database file override previously applied to the

```
BegFunc DeleteOverride Access(*Public) Type(CLProgram)
  DclSrParm fileName Type(*String) Len(45)
  DclSrParm scope Type(ASNA.Monarch.OverrideScope)
```

### Parameters

**fileName**

String containing the database file name to remove the overrides from.

**scope**

**ASNA.Monarch.OverrideScope.** The scope in which the override is to be removed (call or job).

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

CLProgram Class: Members, OverrideFile | ASNA.Monarch Namespace

## InitializePFM Method(string, string, ASNA.Monarch.InitializePFMOptions, integer)

Initializes records in a member of a physical file, by record type, to the total number of records specified.

```
BegFunc InitializePFM Access(Protected) Type(CLProgram)
  DclSrParm File Type(*String) Len(45)
  DclSrParm Mbr Type(*Integer) Len(45)
  DclSrParm Records Type(ASNA.Monarch.InitializePFMOptions)
  DclSrParm TotRcds Type(*Integer4)
```

### Parameters

#### File

String. The qualified name of the physical file containing the member to initialize. The name of the physical file can be qualified by one of the following library values:

- \*LIBL: All libraries in the job's library list are searched until the first match is found. This is the default.
- \*CURLIB: The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.
- library-name: Specify the name of the library to be searched.
- physical-file-name: Specify the name of the physical file.

#### Mbr

String. The optional name of the physical file member within the File to be initialized. The name of the member can be:

- \*FIRST: The first member in the database file is initialized. This is the default.
- \*LAST: The last member of the database file is initialized.
- physical-file-member: Specify the name of the physical file member.

#### Records

ASNA.Monarch.InitializePFMOptions. Specifies the type of records that are initialized and added to the specified member. The records are initialized with Default(0) or Deleted(1) records. If Default is specified, the member is initialized with default record values. Numeric fields are initialized to zeros and all character fields are initialized to blanks. If Deleted is specified, the specified member is initialized with deleted records. The records are not eligible for access, but simply hold a place in the file. Deleted records can be updated to reuse the deleted space.

#### TotRcds

Integer8. Specifies the total number of records in the member after it is initialized. If the value specified in this parameter causes the size of the file to be larger than the size specified when the file was created, a message is sent to the system operator's message queue (QSYSOPR). The operator can either continue or end the operation. If the number of existing records in the member already meets or is larger than this number, no records are initialized; if the number is less than that specified, enough records are initialized to equal the total records specified.

### Remarks

This method initializes records in a member of a physical file to the specified type of record (either default or deleted records). If the initialized member is empty, records are added and initialized to the specified type; if the member is not empty, records of the specified type are added to the member. As many records are added as is necessary to make the total record count specified.

The following restrictions apply:

- The member may be open for input (read only) while the initialize operation takes place but cannot be open for update, delete, or insert.
- To initialize the member with default records, the user needs object operational authority, object management authority or alter, and add authority for the file in which the member exists and execute authority to the library.
- To initialize the member with deleted records, the user also needs delete authority for the file.
- An \*EXCLRD lock is required on the member to initialize it.

- In multithreaded jobs, this command is not thread safe and fails for Distributed Data Management (DDM) files of type \*SNA.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

CLProgram Class | CLProgram Class Members | ASNA.Monarch Namespace

## InitializePFM Method(string, string, integer)

Initializes records in a member of a physical file with default record values, to the total number of records specified.

```
BegFunc InitializePFM Access(Protected) Type(CLProgram)
  DclSrParm File Type(*String) Len(45)
  DclSrParm Mbr Type(*Integer) Len(45)
  DclSrParm TotRcds Type(*Integer4)
```

## Parameters

### File

String. The qualified name of the physical file containing the member to initialize. The name of the physical file can be qualified by one of the following library values:

- \*LIBL: All libraries in the job's library list are searched until the first match is found. This is the default.
- \*CURLIB: The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.
- library-name: Specify the name of the library to be searched.
- physical-file-name: Specify the name of the physical file.

### Mbr

String. The optional name of the physical file member within the File to be initialized. The name of the member can be:

- \*FIRST: The first member in the database file is initialized. This is the default.
- \*LAST: The last member of the database file is initialized.
- physical-file-member: Specify the name of the physical file member.

### TotRcds

Integer8. Specifies the total number of records in the member after it is initialized. If the value specified in this parameter causes the size of the file to be larger than the size specified when the file was created, a message is sent to the system operator's message queue (QSYSOPR). The operator can either continue or end the operation. If the number of existing records in the member already meets or is larger than this number, no records are initialized; if the number is less than that specified, enough records are initialized to equal the total records specified.

## Remarks

This method initializes records in a member of a physical file to default record values. Numeric fields are initialized to zeros and all character fields are initialized to blanks. If the initialized member is empty, records are added and initialized; if the member is not empty, records are added to the member. As many records are added as is necessary to make the total record count specified.

The following restrictions apply:

- The member may be open for input (read only) while the initialize operation takes place but cannot be open for update, delete, or insert.

- To initialize the member with default records, the user needs object operational authority, object management authority or alter, and add authority for the file in which the member exists and execute authority to the library.
- To initialize the member with deleted records, the user also needs delete authority for the file.
- An \*EXCLRD lock is required on the member to initialize it.
- In multithreaded jobs, this command is not thread safe and fails for Distributed Data Management (DDM) files of type \*SNA.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class](#) | [CLProgram Class Members](#) | [ASNA.Monarch Namespace](#)

---

## OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object)

This method provides CLProgram file override options within Monarch.

```
BegFunc OverrideFile Access(*Public) Type(CLProgram)
  DclSrParm fileName Type(*String)
  DclSrParm option Type(ASNA.Monarch.OverrideOptions)
  DclSrParm newValue Type(*object)
```

### Parameters

#### fileName

String containing the database file name to be overridden.

#### option

**ASNA.Monarch.OverrideOptions.** The option being overridden as noted in the enumeration.

#### newValue

Object. The new value to override the existing option value. The value type must be as shown in the OverrideOptions enumeration Table.

### Remarks

See the enumeration for details on the **OverrideOptions**, the value type, and the valid option values for the different types of files.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class: Members, DeleteOverride](#) | [ASNA.Monarch Namespace](#)

## OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object, ASNA.Monarch.OverrideScope)

This method provides CLProgram file override options within Monarch for the specified level of scope.

```
BegFunc OverrideFile Access(*Public)
  Type(CLProgram) DclSrParm fileName Type(*String)
  DclSrParm option Type(ASNA.Monarch.OverrideOptions)
  DclSrParm newValue Type(*object)
  DclSrParm scope Type(ASNA.Monarch.OverrideScope)
```

### Parameters

#### fileName

String containing the database file name to be overridden.

#### option

**ASNA.Monarch.OverrideOptions.** The option being overridden as noted in the enumeration.

#### newValue

Object. The new value to override the existing option value. The value type must be as indicated in the OverrideOptions enumeration table.

#### scope

**ASNA.Monarch.OverrideScope.** The level the override is being applied to (Job or CLProgram).

### Remarks

See the enumeration table for details on the **OverrideOptions**, the value type, and the valid option values for the different types of files.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class: Members, DeleteOverride | ASNA.Monarch Namespace

## Percent\_SST Method

Returns a string containing a substring from the contents of the specified CL character variable or the local data area. This can only be used within a CL procedure.

```
BegFunc Percent_SST Access(*Protected) Type(Void)
  DclSrParm receiver Type (*String) By (*Reference)
  DclSrParm start Type (*Integer)
  DclSrParm length Type (*Integer)
  DclSrParm value Type (*String)
```

### Parameters

#### receiver

String. The string from which a portion is to be returned.

#### start

Integer. Indicates the starting position within the *receiver* string.

#### length

Integer. Indicates the length.

**value**

String. The value to which the return variable is to be changed.

**Returns**

String variable containing a portion of the receiver string with the value changed as specified by the *start* position and *length*.

**Requirements**

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**See Also**

CLProgram Class | CLProgram Class Members | ASNA.Monarch Namespace

---

**Percent\_Switch Method**

This method compares a bit-mask against the current job switch (*sws*) settings and returns a value of '1' if the comparison is equal, otherwise, '0'.

```
BegSr Percent_Switch Access(*Protected) Type(*String) DclSrParm  
mask Type(*String) Len(8)
```

**Parameters****mask**

String. The mask to indicate which job switches are to be compared and the on/off value each is to be tested for. Each position within the masks corresponds with a job switch i.e.: position 1 compared to switch 1. Values for each position within the mask are:

- 0 - Check job switch for a 0 (off).
- 1 - Check job switch for a 1 (on).
- X - Do not check this switch.

**Returns**

String. A value of '1' if the comparison of every switch is the same, otherwise, the result is '0'.

**Requirements**

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**See Also**

CLProgram Class | CLProgram Class Members | ASNA.Monarch Namespace

---

## RetrieveDataArea Method(string)

Returns the data area associated with the job.

```
BegSr RetrieveDataArea Access(*Protected) Type(*String)
DclSrParm dataArea Type(*String) Len(1024)
```

### Parameters

**dataArea**

String. The current program data area name.

### Returns

String. Variable containing the contents of the *dataArea*.

### Remarks

If you want to retrieve many subfields, use this method and then use Percent\_SST to extract subfields.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class](#) | [CLProgram Class Members](#) | [Percent\\_SST Method](#) | [ASNA.Monarch Namespace](#)

---

## RetrieveDataArea Method (string,integer,integer)

Retrieves selected portion of the data area associated with the job.

```
BegSr RetrieveDataArea Access(*Protected) Type(*String)
DclSrParm dataArea Type(*String) Len(1024)
DclSrParm start Type(*Integer)
DclSrParm length Type(*Integer)
```

### Parameters

**dataArea**

String. The current program data area name.

**start**

Integer. The starting point within the data area to be retrieved.

**length**

Integer. The length of the data area to be retrieved.

### Returns

String. Variable containing the partial contents from *dataArea* as specified by the *start* position and *length*.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class](#) | [CLProgram Class Members](#) | [ASNA.Monarch Namespace](#)

## RmvLibEntry Method

This method removes a library name from the user portion of the library list.

```
BegFunc RmvLibEntry Access(Protected) Type(CLProgram)
DclSrParm libraryName Type(*String) Len(45)
```

### Parameters

**libraryName**

String. Specifies the name of the library being removed from the user portion of the library list.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class: Members, AddLibEntry | ASNA.Monarch Namespace

---

## RTVJOBA\_DATE Method

Returns the date from the job attributes for the job in which this CLProgram is used.

```
BegSr RTVJOBA_DATE Access(*Public) Type(CLProgram)
DclSrParm date Type(*Char) Len(6)
```

### Returns

**date**

A character variable containing the date assigned to the job by the system when the job was started. The date returned will be in the date format for the job.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class | CLProgram Members | ASNA.Monarch Namespace

---

## RTVJOBA\_JOB Method

Returns the job name from the job attributes for the job in which this CLProgram is used.

```
BegFunc RTVJOBA_JOB Access(*Public) Type(CLProgram)
DclSrParm job Type(*Char) Len(10)
```

### Returns

**job**

A character variable containing the name of the job. The variable must be at least 10 characters long. If the variable length is longer than the job name, it will be padded on the right with blanks.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class | CLProgram Members | ASNA.Monarch Namespace

---

## RTVJOBA\_NBR Method

Returns the job number from the job attributes for the job in which this CLProgram is used.

```
BegFunc RTVJOBA_NBR Access(*Public) Type(CLProgram)
  DclSrParm nbr Type(*Char) Len(6)
```

### Returns

**nbr**

A character variable containing the unique 6-character number assigned to the job by the system. The job number is the first part of the qualified job name.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class | CLProgram Members | ASNA.Monarch Namespace

---

## RTVJOBA\_SWS Method

Returns the eight switches from the job attributes for the job in which this CLProgram is used.

```
BegFunc RTVJOBA_SWS Access(*Public) Type(void)
  DclSrParm sws Type(*Char) Len(8)
```

### Returns

**sws**

A character variable containing the status value of the eight switches used by the job. The length of the variable must be a minimum of 8 characters. Each position within sws corresponds to a given switch i.e.: position 1 is switch 1. Status values are:

- '0' - Job switch is off.
- '1' - Job switch is on.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

CLProgram Class | CLProgram Members | Job.GetSwitch Method | Job.GetSwitches Method | Job.SetSwitch Method | ASNA.Monarch Namespace

## RTVJOBA\_TYPE Method

Returns the environment type from the job attributes for the job in which this CLProgram is used.

```
BegFunc RTVJOBA_TYPE Access(*Public) Type(CLProgram)
  DclSrParm type Type(*Char) Len(1)
```

### Returns

#### type

A one character variable containing the value representing the environment of the job. Environment values are:

- 0 - Batch job.
- 1 - Interactive job.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class](#) | [CLProgram Members](#) | [ASNA.Monarch Namespace](#)

---

## RTVJOBA\_USER Method

Returns the user profile name from the job attributes for the job in which this CLProgram is used.

```
BegFunc RTVJOBA_USER Access(*Public) Type(CLProgram)
  DclSrParm user Type(*Char) Len(10)
```

### Returns

#### user

A character variable containing the user profile name associated with the job when the job was started. The user name is the second part of the qualified job name. The variable must be a minimum of 10 characters. If the user profile name is less than allowed by the length of the variable, the return value will be padded on the right with blanks.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[CLProgram Class](#) | [CLProgram Members](#) | [ASNA.Monarch Namespace](#)

## Job Class

The **Job** class provides an environment to submit, control, and keep track of program activations.

For a list of all members of this type, see Job Members.

ASNA.Monarch  
**ASNA.Monarch.Job**

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

There is a Monarch Job associated with each user in an ASP.Net application. Actually there is a job with each session kept by ASP.Net. As part of the session startup, a new job is created, a thread is allocated and the Job is started in this thread.

### Job Control

PeekInvocation, PopInvocation, PushInvocation methods

FindProgram, The RegisterProgram method creates an instance of an ASNA.Monarch.Program in the system registry for the specified program, while the UnregisterProgram removes the program from the system registry. The EndPrograms method terminates the current job programs. The ShutDown method terminates the entire session.

### Database Files, File Overrides, and Message Queues

The getDatabase method establishes the database. The Database property is used to get the database associated with the job.

You can use an override command to replace the database file named in a program or to change certain parameters of the existing database files. These overrides can be applied to DBFile, PrintFile, or a WorkStnFile object (ApplyOverrides methods). When you override to a different file(ToFile), the overriding file must have only one record format. A logical file which has multiple record formats defined in DDS may be used if it is defined over only one physical file member. A logical file which has only one record format defined in the DDS may be defined over more than one physical file member. The name of the format does not have to be the same as the format name referred to when the program was created. You should ensure that the format of the data in the overriding file is the same as in the original file, otherwise you may get unexpected results. OverrideFile is an overloaded method that provides file overrides with several parameters options. The DeleteOverride overloaded method is used to remove database file overrides previously applied.

A message is a communication sent from one user, program, or procedure to another. There are two types of messages. Immediate and Predefined. Immediate messages are created by the program or system user when they are sent and are not permanently stored in the system. Predefined messages are created before they are used. These messages are placed in a message file (MessageFileFolder) when they are created, and retrieved from that file when they are used. Messages are sent to and received from message queues, which are separate objects on the system. When a message is sent to a procedure, a program, or a system user, it is placed on a message queue associated with that procedure, program, or user. The procedure, program, or user sees the message by receiving it from the queue (GetInvokedMessageQueue). A message sent to a queue can remain on the queue until it is explicitly received by a program or work station user. Additional fields in support of message files and message queue are ExternalQueue and MessageFileFolder.

### Job /Program Status

The CurrentJob property returns the name of the current active job and the StartupMoment property returns the System.DateTime the job was started.

A program status data structure (Psds) can be defined to make program exception/error information available to a program. There is only one PSDS per module. PsdsJobName, PsdsJobNumber, and PsdsJobUser fields set or return the job name, job number, or user name in the program status data structure, respectively.

### Job Attributes and Variables

Associated with each job on the iSeries there is a set of properties or attributes. These attributes can be programmatically accessed in CL via the RTVJOB commands. One of these attributes is a series of 8 switches which can be 'on' or 'off'. RPG programs also have access to these switches via the \*INUX

indicators. In order to support applications using these switches, this class provides a set of methods to access the 8 values. These are `GetSwitch`, `GetSwitches`, and `SetSwitch`.

A local data area is created for each job in the system. The system creates a local data area, which is initially filled with blanks, with a length of 1024 and type `*CHAR`. When you submit a job, the value of the submitting job's local data area is copied into the submitted job's local data area. You can use this local data area to pass information to a procedure or program without the use of a parameter list. The `SetLdaField` and `GetLdaField` methods set or return a specified portion of the `Lda` respectively. The protected `Lda` field is a reference to the `System.Text.StringBuilder` object with the contents of the `Lda`.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA.VisualRPG.Runtime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Job Members](#) | [WebJob Class](#) | [ASNA.Monarch Namespace](#)

---

## Job Constructor()

Creates a new instance of an `ASNA.Monarch.Job` object.

```
BegFunc Job() Access(*Public) Type(Job)
```

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Job Class](#) | [Job Class Members](#) | [ASNA.Monarch Namespace](#)

---

## ApplyOverrides Method (string, ASNA.VisualRPG.Runtime.DBfile)

This method overrides the database file currently specified for the job.

```
BegFunc ApplyOverrides Access(*Public) Type(Job)
  DclSrParm fileName Type(*String) Len(45)
  DclSrParm file      Type(ASNA.VisualRPG.Runtime.DBfile)
```

## Parameters

### fileName

String containing the database file name to override.

### file

`ASNA.VisualRPG.Runtime.DBfile`. The file to be used in place of the *fileName* for the job.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Job Class Members](#) | [ASNA.Monarch Namespace](#)

## ApplyOverrides Method (string, ASNA.VisualRPG.Runtime.PrintFile)

This method overrides the print file currently specified for the job.

```
BegFunc ApplyOverrides Access(*Public) Type(Job)
  DclSrParm fileName Type(*String) Len(45)
  DclSrParm file      Type(ASNA.VisualRPG.Runtime.PrintFile)
```

### Parameters

#### fileName

String containing the print file name to override.

#### file

ASNA.VisualRPG.Runtime.PrintFile. The file to be used in place of the *fileName* for the job.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

Job Class Members | ASNA.Monarch Namespace

## ApplyOverrides Method (string, ASNA.VisualRPG.Runtime.WorkStnFile)

This method overrides the workstation file currently specified for the job.

```
BegFunc ApplyOverrides Access(*Public) Type(Job)
  DclSrParm fileName Type(*String) Len(45)
  DclSrParm file      Type(ASNA.VisualRPG.Runtime.WorkStnFile)
```

### Parameters

#### fileName

String containing the workstation file name to override.

#### file

ASNA.VisualRPG.Runtime.WorkStnFile. The file to be used in place of the *fileName* for the job.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

Job Class Members | ASNA.Monarch Namespace

## DeleteOverride Method(string)

This method removes the database file override previously applied to the job.

```
BegFunc DeleteOverride Access(*Public) Type(Job)
  DclSrParm fileName Type(*String) Len(45)
```

## Parameters

### fileName

String containing the database file name to remove overrides from.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | ASNA.Monarch Namespace

---

## DeleteOverride Method(string, ASNA.Monarch.OverrideScope)

This method removes the database file override previously applied to the job.

```
BegFunc DeleteOverride Access(*Public) Type(Job)
  DclSrParm fileName Type(*String) Len(45)
  DclSrParm scope Type(ASNA.Monarch.OverrideScope)
```

## Parameters

### fileName

String containing the database file name to remove the overrides from.

### scope

ASNA.Monarch.OverrideScope. The scope at which the override is to be removed (call or job).

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class Members | ASNA.Monarch Namespace | OverrideScope Enumeration

---

## DeleteOverride Method(string, integer)

This method removes the database file override previously applied to the job on either a call or job level.

```
BegFunc DeleteOverride Access(*Public) Type(Job)
  DclSrParm fileName Type(*String) Len(45)
  DclSrParm level Type(*Integer)
```

## Parameters

### fileName

String containing the database file name to remove the override from.

### level

Integer. The level in which the override is to be removed.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

---

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class Members](#) | [ASNA.Monarch Namespace](#)

---

## EndPrograms Method

This method terminates the **Job** program(s).

```
BegSR EndPrograms Access(*Public) Type(Job)
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [ASNA.Monarch Namespace](#)

---

## FindProgram Method

Returns an ASNA.Monarch.Program object for the program type.

```
BegFunc FindProgram Access(*Public) Type(ASNA.Monarch.Program)  
DclSrParm programType Type(System.Type)
```

### Parameters

**programType**

**System.Type.** The type to obtain the ASNA.Monarch.Program object for.

### Returns

**ASNA.Monarch.Program** object for the *programType*.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [ASNA.Monarch Namespace](#)

---

## getDatabase Method

The getDatabase method creates a new instance of ASNA.VisualRPG.Runtime.Database connection object for the job.

```
BegFunc getDatabase Access(*Protected) Type(ASNA.VisualRPG.Runtime.Database)
```

### Returns

A new instance of ASNA.VisualRPG.Runtime.Database connection for the job.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

Job Class | Job Members | Database Property | ASNA.Monarch Namespace

---

## GetInvokedMessageQueue(int)

Returns an ASNA.Monarch.MessageQueue object by invocation offset.

```
BegFunc GetInvokedMessageQueue Access(*Public) Type(ASNA.Monarch.MessageQueue)
DclSrParm invocationOffset Type(*Integer4)
```

### Parameters

**invocationOffset**

The offset number within the message queue invocation list to be returned.

### Returns

An ASNA.Monarch.MessageQueue object by invocation offset.

### Remarks

To retrieve the ASNA.Monarch.MessageQueue object by program name, use GetInvokedMessageQueue(string).

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

Job Class | Job Members | ASNA.Monarch Namespace

---

## GetInvokedMessageQueue(string)

Returns an ASNA.Monarch.MessageQueue object by program name.

```
BegFunc GetInvokedMessageQueue Access(*Public) Type(ASNA.Monarch.MessageQueue)
DclSrParm programName Type(*String)
```

### Parameters

**programName**

The program name within the message queue invocation list to be returned.

### Returns

An ASNA.Monarch.MessageQueue object for the specified program name.

### Remarks

To retrieve the ASNA.Monarch.MessageQueue object by invocation offset, use GetInvokedMessageQueue(int).

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

---

## See Also

Job Class | Job Members | ASNA.Monarch Namespace

---

## GetLdaField Method

Returns a specified portion of the local data area for the job.

```
BegFunc GetLdaField Access(*Public) Type(*String)
  DclSrParm start Type(*Integer2)
  DclSrParm length Type(*Integer2)
```

### Parameters

**start**

The starting position with the local data area to begin retrieving data.

**length**

The length of the field to retrieve from the local data area.

### Returns

A string containing the portion of the program local data area starting at the *start* position for the *length* specified.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | SetLdaField | ASNA.Monarch Namespace

---

## GetSwitches Method

Returns a string of 8 characters with values '0' and '1', each character represents one of the job attribute switches.

```
BegFunc GetSwitches Access(*Public) Type(*String)
```

### Returns

A string of 8 characters with values '0' and '1'. Each character represents one of the switches. '0' if the switch is 'off' or a '1' if the switch is 'on'.

### Remarks

RPG programs have access to these switches via the \*INUX indicators. In order to support applications using these switches, the **Job** class provides a set of methods to access the 8 values. Additional methods to return or set a specific switch are GetSwitch and SetSwitch respectively. The CLProgram class also provides the method RTVJOBA\_SWS to facilitate the translation of the RTVJOBA CL command SWS attribute.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | ASNA.Monarch Namespace

## GetSwitch Method

Returns a value indicating the 'on/off' status of a specific job attribute switch.

```
BegFunc GetSwitch Access(*Public) Type(*Char)
  DclSrParm iSwitch Type(*Integer2)
```

### Parameters

#### iSwitch

An integer value between 1 and 8 representing the desired switch.

### Returns

A '0' if the switch is 'off' or a '1' if the switch is 'on'.

### Remarks

RPG programs have access to these switches via the \*INUX indicators. In order to support applications using these switches, the **Job** class provides a set of methods to access the 8 values. Additional methods to return or set these switches are GetSwitches and SetSwitch. The CLProgram class also provides the method RTVJOBA\_SWS to facilitate the translation of the RTVJOBA CL command SWS attribute.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

Job Class | Job Members | ASNA.Monarch Namespace

---

## OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object)

This method provides Job file override options within Monarch.

```
BegFunc OverrideFile Access(*Public)
  Type(Job) DclSrParm fileName Type(*String)
  DclSrParm option Type(ASNA.Monarch.OverrideOptions)
  DclSrParm newValue Type(*object)
```

### Parameters

#### fileName

String containing the database file name to be overridden.

#### option

**ASNA.Monarch.OverrideOptions.** The option being overridden as noted in the table below.

#### newValue

Object. The new value to override the existing option value. The value type must be as shown in the **OverrideOptions** enumeration table under the descriptions column.

### Remarks

The **OverrideOptions**, the value type, and the valid option values for the different types of files are shown in the enumeration table.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class Members | ASNA.Monarch Namespace

---

## OverrideFile Method(string, ASNA.Monarch.OverrideOptions, object, ASNA.Monarch.OverrideScope)

This method provides Job file override options within Monarch for the specified level of scope.

```

BegFunc OverrideFile Access(*Public)
    Type(Job) DclSrParm fileName Type(*String)
    DclSrParm option Type(ASNA.Monarch.OverrideOptions)
    DclSrParm newValue Type(*object)
    DclSrParm scope Type(ASNA.Monarch.OverrideScope)

```

### Parameters

**fileName**

String containing the database file name to be overridden.

**option**

**ASNA.Monarch.OverrideOptions.** The option being overridden as noted in the enumeration table.

**newValue**

Object. The new value to override the existing option value. The value type must be as indicated below.

**scope**

**ASNA.Monarch.OverrideScope.** The level the override is being applied to (Job or CLProgram).

### Remarks

The **OverrideOptions**, the value type, and the valid option values for the different types of files are shown in the enumeration table.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class Members | ASNA.Monarch Namespace

---

## PeekInvocation Method(int)

Creates an instance of an ASNA.Monarch.Invocation object by offset from the top.

```

BegFunc PeekInvocation Access(*Public) Type(ASNA.Monarch.Invocation)
    DclSrParm offsetFromTop Type(*Integer4)

```

### Parameters

**offsetFromTop**

The offset number.

### Returns

An ASNA.Monarch.Invocation object for the offset indicated.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [PeekInvocation Method\(str\)](#) | [ASNA.Monarch Namespace](#)

---

## PeekInvocation Method(String)

Creates an instance of an ASNA.Monarch.Invocation object by program name.

```
BegFunc PeekInvocation Access(*Public) Type(ASNA.Monarch.Invocation)
    DclSrParm programName Type(*String)
```

### Parameters

**programName**

The program name.

### Returns

An ASNA.Monarch.Invocation object for the program name.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [PeekInvocation Method\(int\)](#) | [ASNA.Monarch Namespace](#)

---

## PopInvocation Method

Returns the last instance of an ASNA.Monarch.Invocation object in the job stack.

```
BegFunc PopInvocation() Access(*Public) Type(ASNA.Monarch.Invocation)
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [ASNA.Monarch Namespace](#)

---

## PushInvocation Method

Creates an instance of an ASNA.Monarch.Program in the job stack.

```
BegFunc PushInvocation Access(*Public) Type(void)
    DclSrParm program Type(ASNA.Monarch.Program)
```

### Parameters

**program**

ASNA.Monarch.Program.

---

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | ASNA.Monarch Namespace

---

## RegisterProgram Method

Creates an instance of an ASNA.Monarch.Program in the system registry for the specified program.

```
BegFunc RegisterProgram Access(*Public) Type(void)
    DclSrParm program Type(ASNA.Monarch.Program)
```

## Parameters

### program

ASNA.Monarch.Program.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | UnregisterProgram Method | ASNA.Monarch Namespace

---

## SetLdaField Method

The SetLdaField method sets a new value in the specified portion of the local data area for the job.

```
BegFunc SetLdaField Access(*Public) Type(*String)
    DclSrParm start Type(*Integer2)
    DclSrParm length Type(*Integer2)
    DclSrParm newValue Type(*char)
```

## Parameters

### start

The starting position with the local data area that the newValue data will be inserted.

### length

The length of the newValue field to be inserted into the local data area.

### newValue

The new value to be inserted into the local data area at the start position for the length specified.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | GetLdaField | ASNA.Monarch Namespace

## SetSwitch Method

Set a value indicating the 'on/off' status of a specific job attribute switch.

```
BegFunc SetSwitch Access(*Public) Type(void)
  DclSrParm iSwitch Type(*Integer2)
  DclSrParm value Type(*Char)
```

### Parameters

#### iSwitch

An integer value between 1 and 8 representing the desired switch.

#### value

A one character value to indicate the status of iSwitch. Set the value '0' to turn the switch 'off' or '1' to turn the switch 'on'.

### Remarks

RPG programs have access to these switches via the \*INUX indicators. In order to support applications using these switches, the **Job** class provides a set of methods to access the 8 values. Additional methods to return these switches are GetSwitches and GetSwitch. The CLProgram class also provides the method RTVJOBA\_SWS to facilitate the translation of the RTVJOBA CL command SWS attribute.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [ASNA.Monarch Namespace](#)

---

## ShutDown Method

This method terminates the Job program when an abnormal condition occurs.

```
BegFunc ShutDown() Type(void)
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [ASNA.Monarch Namespace](#)

---

## UnregisterProgram Method

Removes the ASNA.Monarch.Program object in the system registry for the specified program.

```
BegFunc UnregisterProgram Access(*Public) Type(void)
  DclSrParm program Type(ASNA.Monarch.Program)
```

### Parameters

#### program

ASNA.Monarch.Program.

---

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | RegisterProgram Method | ASNA.Monarch Namespace

---

## CurrentJob Property

The **CurrentJob** property gets the Job object corresponding to this Job.

```
BegFunc CurrentJob Access(*Protected) Type(ASNA.Monarch.Job)
```

## Returns

ASNA.Monarch.Job.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | ASNA.Monarch Namespace

---

## Database Property

The **Database** property returns the ASNA.VisualRPG.Runtime.Database connection object corresponding to this Job.

```
BegFunc Database Access(*Protected) Type(ASNA.VisualRPG.Runtime.Database)
```

## Returns

ASNA.VisualRPG.Runtime.Database connection object for the job.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | getDatabase Method | ASNA.Monarch Namespace

---

## MessageFileFolder Property

The **MessageFileFolder** property returns or sets the name of the folder containing the message file for this job.

```
BegFunc MessageFileFolder Access(*Public) Type(*String)
```

## Field Value

A string variable containing the name of the fold containing the message file for the job.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | MessageFileFolder Field | ASNA.Monarch Namespace

---

## StartupMoment Property

The **StartupMoment** property returns the System.DateTime when the job was started.

```
BegFunc StartupMoment Access(*Protected) Type(System.DateTime)
```

## Returns

**System.DateTime.**

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | ASNA.Monarch Namespace

---

## ExternalQueue Field

Returns or sets an instance of an ASNA.Monarch.MessageQueue object for an external message queue for this job.

```
BegFunc ExternalQueue Access(*Public) Type(ASNA.Monarch.MessageQueue)
```

## Field Value

**ASNA.Monarch.MessageQueue** referencing the external message queue object for the job.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Job Class | Job Members | ASNA.Monarch Namespace

---

## Ida Field

This protected field returns a reference to the **System.Text.StringBuilder** object with the contents of the local data area for the job.

```
BegFunc Ida Access(*Protected) Type(System.Text.StringBuilder)
```

## Field Value

**System.Text.StringBuilder** object for the local data area for the job.

---

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Job Class](#) | [Job Members](#) | [GetLdaField Method](#) | [ASNA.Monarch Namespace](#)

---

## MessageFileFolder Field

This protected field returns a reference to the message file folder object containing the message file for this job.

```
BegFunc MessageFileFolder Access(*Protected) Type(*String)
```

## Field Value

The string referencing the message file folder object for the message file used by the job.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Job Class](#) | [Job Members](#) | [MessageFileFolder Property](#) | [ASNA.Monarch Namespace](#)

---

## PsdsJobName Field

Returns or sets the 10 character job name field in the program status data structure.

```
BegFunc PsdsJobName Access(*Public) Type(*String) Len(10)
```

## Field Value

Ten character string containing the job name.

## Remarks

A program status data structure (PSDS) can be defined to make program exception/error information available to a program. There is only one PSDS per module. The job name is in position 244-253 of the data structure.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Job Class](#) | [Job Members](#) | [GetLdaField Method](#) | [ASNA.Monarch Namespace](#)

---

## PsdsJobNumber Field

Returns or sets the 6 digits job number field in the program status data structure.

```
BegFunc PsdsJobNumber Access(*Public) Type(*Decimal) Len(6,0)
```

### Field Value

Six digit numeric containing the job number.

### Remarks

A program status data structure (PSDS) can be defined to make program exception/error information available to a program. There is only one PSDS per module. The job number is in position 264-269 of the data structure.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [GetLdaField Method](#) | [ASNA.Monarch Namespace](#)

---

## PsdsJobUser Field

Returns or sets the 10 character user name field in the program status data structure.

```
BegFunc PsdsJobUser Access(*Public) Type(*String) Len(10)
```

### Field Value

Ten character string containing the user name from the user profile.

### Remarks

A program status data structure (PSDS) can be defined to make program exception/error information available to a program. There is only one PSDS per module. The user name is in position 254-263 of the data structure.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Job Class](#) | [Job Members](#) | [GetLdaField Method](#) | [ASNA.Monarch Namespace](#)

## Program Class

The **Program** class provides activation/invoke semantics. It conforms to certain conventions allowing it to be the Target of a Call operation.

The class must implement the **\*Entry shared** subroutine.

This class provides support in the areas of:

- Local Data Area(Lda)
- Message Queues
- Program Control.

For a list of all members of this type, see Program Class Members.

ASNA.Monarch

### ASNA.Monarch.Program

```
BegClass Program Access(public) Inherits (System.Object)
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

### Local Data Area (Lda)

A local data area is created for each job in the system. The system creates a local data area, which is initially filled with blanks, with a length of 1024 and type \*CHAR. When you submit a job, the value of the submitting job's local data area is copied into the submitted job's local data area. You can use this local data area to pass information to a procedure or program without the use of a parameter list. The DataArea\_In method returns the input data area for the program with the specified lock option. The DataArea\_Out method updates the data area input with new values, also updating the lock option. The DataArea\_Unlock method unlocks a previously locked data area. The SetLdaField and GetLdaField methods set or return a specified portion of the Lda respectively.

### Message Queues

A message is a communication sent from one user, program, or procedure to another. There are two types of messages. Immediate and Predefined. Immediate messages are created by the program or system user when they are sent and are not permanently stored in the system. Predefined messages are created before they are used. These messages are placed in a message file when they are created, and retrieved from that file when they are used. Messages are sent to and received from message queues, which are separate objects on the system. When a message is sent to a procedure, a program, or a system user, it is placed on a message queue associated with that procedure, program, or user. The SendExternalMessage method sends an immediate or predefined external program message and returns a response depending on the type of message sent. The response may be entered by the user, the default message (if any), or \*N. The overloaded SendProgramMessage method sends an immediate or predefined program message. The different overloaded methods allow for messages to be sent to specific message files and program queues and to return a response depending on the type of message sent. The overloaded RemoveMessage method removes a message by message key or message key and program queue.

### Program Control

The ExecCommand method executes the program command specified. The monarchJob read-only field contains the name of the ASNA.Monarch.Job under which the Program is running. The Dispose returns true if the program is in the process of being disposed of. In addition, UDATE, UDAY, UMONTH, UYEAR read-only properties return the program run-time date, day, month, and year respectively.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA.VisualRPG.Runtime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

ASNA.Monarch Namespace | Program Class Members | ASNA.Monarch.MessageTypes

---

## Program Constructor

```
BegSr Program Access(*Public) Type(Program)
```

## Returns

A new instance of a **Program** object.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Program Class | Program Class Members | ASNA.Monarch Namespace

---

## DataArea\_In Method

Returns the data area as input to the program.

```
BegFunc DataArea_In Access(*Protected) Type(*String)
  DclSrParm library Type(*String)
  DclSrParm dataarea Type(*String)
  DclSrParm withLock Type(*Boolean)
```

## Parameters

### library

String. The name of the library.

### dataarea

String. The name of dataarea.

### withLock

Boolean. Set **True** if the data area is locked, otherwise **False**.

## Returns

String. The data area from the library with the lock option specified.

## Remarks

See DataArea\_out.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

Program Class | Program Class Members | ASNA.Monarch Namespace

---

## DataArea\_Out Method

Creates a new instance of a data area.

```
BegSr DataArea_Out Access(*Protected) Type(Program)
  DclSrParm library Type(*String)
  DclSrParm dataarea Type(*String)
  DclSrParm newValue Type(*String)
  DclSrParm keepLock Type(*Boolean)
```

### Parameters

**library**

String. The name of the library.

**dataarea**

String. The name of dataarea.

**newValue**

String. The new data area values for the substitution variables in the dataarea.

**keepLock**

Boolean. Set **True** to indicate the new dataarea is to be locked, otherwise set **False**.

### Returns

A new instance of a data area with the *newValues* substituted within the variables.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## DataArea\_Unlock Method

Unlocks a library data area.

```
BegSr DataArea_Unlock Access(*Protected) Type(Program)
  DclSrParm library Type(*String)
  DclSrParm dataarea Type(*String)
```

### Parameters

**library**

String. The name of the library in which the dataarea is located.

**dataarea**

String. The name of data area to unlock.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

## Dispose Method

Releases resources used by the **WebDevice** object.

```
BegSr Dispose Access(*Public) Type(void)
DclSrParm disposing Type(*Boolean)
```

### Parameters

#### disposing

Boolean. Set **true** to release both managed and unmanaged resources; otherwise set **false** to release just unmanaged resources.

### Remarks

When the **Program** is being disposed of, it can no longer be referenced. Even though the instance of a **Program** object is disposed of, it is still in memory until removed through garbage collection.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## ExecCommand Method

Executes a program command.

```
BegSr ExecCommand Access(*Protected) Type(void)
DclSrParm cmdText Type(*String)
```

### Parameters

#### cmdText

String. This string contains the command to be executed and additional parameters, text, elements, qualifiers, etc., as needed for the command definition.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## GetLdaField Method

Returns a specified portion of the local data area for the program.

```
BegFunc GetLdaField Access(*Public) Type(*String)
DclSrParm start Type(*Integer2)
DclSrParm length Type(*Integer2)
```

---

## Parameters

**start**

The starting position with the local data area to begin retrieving data.

**length**

The length of the field to retrieve from the local data area.

## Returns

A string containing the portion of the program local data area starting at the *start* position for the *length* specified.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Members](#) | [SetLdaField](#) | [ASNA.Monarch Namespace](#)

---

## RemoveMessage Method(string)

Remove a program message.

```
BegFunc RemoveMessage Access(*Protected) Type(Program)
DclSrParm messageKey Type(*String)
```

## Parameters

**messageKey**

String. The unique message identification key.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## RemoveMessage Method(string, string)

Remove a message from the program queue.

```
BegFunc RemoveMessage Access(*Protected) Type(Program)
DclSrParm pgmQ Type(*String)
DclSrParm messageKey Type(*String)
```

## Parameters

**pgmQ**

String. The name of the program queue from which the message is to be removed.

**messageKey**

String. The unique message identification key.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## SendExternalMessage Method(string, string, string, ASNA.Monarch.MessageTypes)

Send an external program message.

```
BegFunc SendExternalMessage Access(*Protected) Type(*String)
    DclSrParm id Type(*String)
    DclSrParm file Type(*String)
    DclSrParm data Type(*String)
    DclSrParm type Type(ASNA.Monarch.MessageTypes)
```

## Parameters

### id

String. The name of the message identifier for predefined messages, or an immediate message.

### file

String. The name of the message file containing the message description for predefined messages.

### data

String. For predefined messages this contains values for the substitution variables in the message. The format of each variable must be defined in the message description. If an immediate message is being sent, there are no data fields.

### type

**ASNA.Monarch.MessageTypes.** Specifies the type of message being sent. i.e.: Completion, Diagnostic, Escape, Informational, Inquiry, Notify, Request, or Status.

## Returns

String. Depending on the *type* of message sent. Any response entered by the user, the default message (if any), or \*N.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch.MessageTypes](#) | [ASNA.Monarch Namespace](#)

---

## SendProgramMessage Method(string)

Send a program message.

```
BegFunc SendProgramMessage Access(*Protected) Type(*String)
    DclSrParm text Type(*String)
```

---

## Parameters

### text

String. Specifies the message text.

## Returns

String. Any response entered by the user, the default message (if any), or \*N.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## SendProgramMessage Method(string, string, ASNA.Monarch.MessageTypes)

Send a program message.

```
BegFunc SendProgramMessage Access(*Protected) Type(*String)
  DclSrParm text Type(*String)
  DclSrParm pgmQ Type(*String)
  DclSrParm type Type(ASNA.Monarch.MessageTypes)
```

## Parameters

### text

String. The message text.

### pgmQ

String. The name of the program queue to send the message to.

### type

**ASNA.Monarch.MessageTypes.** Specifies the type of message being sent. i.e.: Completion, Diagnostic, Escape, Informational, Inquiry, Notify, Request, or Status.

## Returns

String. Depending on the *type* of message sent. Any response entered by the user, the default message (if any), or \*N.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch.MessageTypes](#) | [ASNA.Monarch Namespace](#)

## SendProgramMessage Method(string, string, string)

Send a program message.

```
BegFunc SendProgramMessage Access(*Protected) Type(*String)
  DclSrParm id Type(*String)
  DclSrParm file Type(*String)
  DclSrParm data Type(*String)
```

### Parameters

**id**

String. The name of the message identifier for predefined messages, or an immediate message.

**file**

String. The name of the message file containing the message description for predefined messages.

**data**

String. For predefined messages this contains values for the substitution variables in the message. The format of each variable must be defined in the message description. If an immediate message is being sent, there are no data fields.

### Returns

String. Depending on the *type* of message sent. Any response entered by the user, the default message (if any), or \*N.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## SendProgramMessage Method(string, string, string, string)

Send a program message.

```
BegFunc SendProgramMessage Access(*Protected) Type(*String)
  DclSrParm id Type(*String)
  DclSrParm file Type(*String)
  DclSrParm data Type(*String)
  DclSrParm pgmQ Type(*String)
```

### Parameters

**id**

String. The name of the message identifier for predefined messages, or an immediate message.

**file**

String. The name of the message file containing the message description for predefined messages.

**data**

String. For predefined messages this contains values for the substitution variables in the message. The format of each variable must be defined in the message description. If an immediate message is being sent, there are no data fields.

**pgmQ**

String. The name of the program queue to send the message to.

## Returns

String. Depending on the *type* of message sent. Any response entered by the user, the default message (if any), or \*N.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## SendProgramMessage Method(string, string, string, string, ASNA.Monarch.MessageTypes)

Send a program message.

```
BegFunc SendProgramMessage Access(*Protected) Type(*String)
    DclSrParm id Type(*String)
    DclSrParm file Type(*String)
    DclSrParm data Type(*String)
    DclSrParm pgmQ Type(*String)
    DclSrParm type Type(ASNA.Monarch.MessageTypes)
```

## Parameters

### id

String. The name of the message identifier for predefined messages, or an immediate message.

### file

String. The name of the message file containing the message description for predefined messages.

### data

String. For predefined messages this contains values for the substitution variables in the message. The format of each variable must be defined in the message description. If an immediate message is being sent, there are no data fields.

### pgmQ

String. The name of the program queue to send the message to.

### type

**ASNA.Monarch.MessageTypes.** Specifies the type of message being sent. i.e.: Completion, Diagnostic, Escape, Informational, Inquiry, Notify, Request, or Status.

## Returns

String. Depending on the *type* of message sent. Any response entered by the user, the default message (if any), or \*N.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch.MessageTypes](#) | [ASNA.Monarch Namespace](#)

## SetLdaField Method

The SetLdaField method sets new values in the specified portion of the local data area for the program.

```
BegFunc SetLdaField Access(*Protected) Type(*String)
  DclSrParm start Type(*Integer2)
  DclSrParm length Type(*Integer2)
  DclSrParm newValue Type(*char)
```

### Parameters

#### start

The starting position within the local data area that the newValue data will be inserted.

#### length

The length of the newValue field to be inserted into the local data area.

#### newValue

The new value to be inserted into the local data area at the start position for the length specified.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Members](#) | [GetLdaField](#) | [ASNA.Monarch Namespace](#)

---

## UPDATE Property

Get the Program run-time date.

```
BegFunc UPDATE Access(*Protected) Type(*String) Len(6)
```

### Returns

Read-only. String containing the current run-time date.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## UDAY Property

Get the Program run-time day.

```
BegFunc UDAY Access(*Protected) Type(*String) Len(2)
```

### Returns

Read-only. String containing the current run-time day.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## UMONTH Property

Get the Program run-time month.

```
BegFunc UMONTH Access(*Protected) Type(*String) Len(2)
```

### Returns

Read-only. String containing the current run-time month.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## UYEAR Property

Get the Program run-time year.

```
BegFunc UYEAR Access(*Protected) Type(*String) Len(2)
```

### Returns

Read-only. String containing the current run-time year.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

---

## monarchJob Field

The name of the **ASNA.Monarch.Job** under which the **Program** is running.

```
BegSr monarchJob Access(*Protected) Type(ASNA.Monarch.Job)
```

### Returns

The name of the **ASNA.Monarch.Job** under which the Program is running.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**See Also**

[Program Class](#) | [Program Class Members](#) | [ASNA.Monarch Namespace](#)

## WebDevice Class

The **WebDevice** class provides for device control for web display, program, and job objects.

For a list of all members of this type, see [WebDevice Members](#).

ASNA.Monarch

### ASNA.Monarch.WebDevice

```
public class WebDevice Inherits System.Object
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

The WebDevice class provides the device controls.

- **WebDevice** creates an instance of a WebDevice object for a WebJob object.
- **Attach** attaches a WebDisplayFile file to a device.
- **Detach** is used to remove the device from a file.
- **SignalDataReadyForProgram** is used to indicate the device is ready for data from a program event.
- **SignalDataReadyForUser** is used to indicate the device is ready for data from a user event.
- **WaitForDataForProgram** is used to indicate the device is to wait for data from a program event.
- **WaitForDataForUser** is used to indicate the device is to wait for data from a user event.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA.VisualRPG.Runtime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ASNA.Monarch Namespace | WebDevice Class Members](#)

---

## WebDevice (ASNA.Monarch.WebJob)

Initialize a new instance of a **WebDevice** object for a WebJob

```
BegSr WebDevice Access(*Public) Type(WebDevice)
  DclSrParm job Type(ASNA.Monarch.WebJob)
```

## Parameters

**job**

**ASNA.Monarch.WebJob.** The name of the WebJob.

## Returns

A new instance of a **WebDevice** object for the **WebJob**.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[Program Class](#) | [Program Class Members](#) | [WebJob Class](#) | [ASNA.Monarch Namespace](#)

---

## Attach Method(string, ASNA.Monarch.WebDisplayFile)

Returns or sets the ASNA.Monarch.WebDisplayFile for a **WebDevice** object and also attaches a shared file.

```
BegSr Attach Access(*Public) Type(WebDevice)
    DclSrParm shareFileName Type(*String)
    DclSrParm displayFile Type(ASNA.Monarch.WebDisplayFile)
```

## Parameters

### shareFileName

The shared file name to be attached to the device.

### displayFile

ASNA.Monarch.WebDisplayFile. The WebDisplayFile object to be attached to the **WebDevice**.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [Attach \(ASNA.Monarch.WebDisplayFile\)](#) | [WebDisplayFile Class](#) | [ASNA.Monarch Namespace](#)

---

## Attach Method(ASNA.Monarch.WebDisplayFile)

Returns or sets the ASNA.Monarch.WebDisplayFile for a WebDevice object.

```
BegSr Attach Access(*Public) Type(WebDevice)
    DclSrParm displayFile Type(ASNA.Monarch.WebDisplayFile)
```

## Parameters

### displayFile

ASNA.Monarch.WebDisplayFile. The WebDisplayFile object to be attached to the **WebDevice**.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [Attach \(string, ASNA.Monarch.WebDisplayFile\)](#) | [WebDisplayFile Class](#) | [ASNA.Monarch Namespace](#)

---

## Detach Method(string)

Detaches a previously attached shared file from the **WebDevice**.

```
BegSr Detach Access(*Public) Type(WebDevice)
DclSrParm shareFileName Type(*String)
```

### Parameters

**shareFileName**

The shared file name to be detached from the **WebDevice**.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

WebDevice Class Members | ASNA.Monarch Namespace

---

## Detach Method(ASNA.Monarch.WebDisplayFile)

Detaches a previously attached ASNA.Monarch.WebDisplayFile from a **WebDevice** object.

```
BegSr Detach Access(*Public) Type(WebDevice)
DclSrParm displayFile Type(ASNA.Monarch.WebDisplayFile)
```

### Parameters

**displayFile**

ASNA.Monarch.WebDisplayFile. The **WebDisplayFile** object to be detached from the WebDevice

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

WebDevice Class: Members, Attach, Detach (string) | WebDisplayFile Class | ASNA.Monarch Namespace

---

## Dispose Method

Releases resources used by the **WebDevice** object.

```
BegSr Dispose Access(*Public) Type(WebDevice)
```

### Remarks

Release both managed and unmanaged resources used by the **WebDevice**.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

---

## GetActiveFile Method

Returns an instance of the **WebDisplayFile** object for the **WebDevice**.

```
BegSr GetActiveFile Access(*Public) Type(WebDisplayFile)
```

## Returns

An instance of the **WebDisplayFile** object.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [WebDisplayFile Class](#) | [ASNA.Monarch Namespace](#)

---

## GetThreadDevice Method

Returns an instance of a **WebDevice** thread object.

```
BegSr GetThreadDevice Access(*Public) Type(WebDevice)
```

## Returns

An instance of the **WebDevice** thread object containing application instructions.

## Remarks

A thread can execute a part of an application's code, including code that is currently being executed by another thread. Threads share the virtual address space, global variables, and operating-system resources of their respective processes.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [WebDevice Class](#) | [ASNA.Monarch Namespace](#)

---

## Read Method(string)

This read method will be removed in release 3. Use [Read \(WebDisplayFile dspFile\)](#)

```
BegSr Read Access(*Public) Type(WebDevice)  
DclSrParm dspFileName Type(*String)
```

## Parameters

### dspFileName

The file name of the web display file to read.

## Returns

The **WebDevice** object for the display file named.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

---

## Read Method(ASNA.Monarch.WebDisplayFile)

Read the specified WebDisplayFile object for the WebDevice.

```
BegSr Read Access(*Public) Type(WebDevice)
    DclSrParm dspFile Type(ASNA.Monarch.WebDisplayFile)
```

## Parameters

### dspFile

ASNA.Monarch.WebDisplayFile. The file name of the web display file to return.

## Returns

The **WebDevice** object for the display file named.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch.WebDisplayFile](#) | [ASNA.Monarch Namespace](#)

---

## SignalDataReadyForProgram Method

Signals the **WebDevice** object thread the device is ready for data from a program event.

```
BegSub SignalDataReadyForProgram Access(*Public)
```

## Remarks

This method is used to signal to the **WebDevice** object program thread that that device is ready the data from a program event. The thread state of the current program will be changed from "waiting" to "running" or "ready". Use **WaitForDataForProgram** to place the thread in the 'wait' state.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

## SignalDataReadyForUser Method

Signals the **WebDevice** object thread indicate the device is ready for data from a user event.

```
BegSub SignalDataReadyForUser Access(*Public)
```

### Remarks

This method is used to signal to the **WebDevice** object program thread that the device is ready for data from a user event. The thread state of the current program will be changed from "waiting" to "running" or "ready". Use **WaitForDataForUser\_** to place the thread in the 'wait' state.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

---

## WaitForDataForProgram Method

Signals the **WebDevice** object thread the device is to wait for data from a program event.

```
BegSub WaitForDataForProgram Access(*Public)
```

### Remarks

This method is used to signal to the **WebDevice** object the device is to wait for data from a program event. The thread state of the current program will be changed from "running" or "ready" to "waiting".

This **WaitForDataForProgram** method returns when the WebDevice object is in the signaled state, or when the time-out interval elapses. This function does not return if there is unread input in the queue. It returns only when new input arrives. Use **SignalDataReadyForProgram** when the data is available and to remove the wait state.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

---

## WaitForDataForUser Method

Signals the **WebDevice** object thread the device is to wait for data from a user event.

```
BegSub WaitForDataForUser Access(*Public)
```

### Remarks

This method is used to signal to the **WebDevice** object program thread to wait for data from a user event. The thread state of the current program will be changed from "running" or "ready" to "waiting".

This **WaitForDataForUser** method returns when the WebDevice object is in the signaled state, or when the time-out interval elapses. This function does not return if there is unread input in the queue. It returns only when new input arrives. Use **SignalDataReadyForUser\_** when the user input data is available and to remove the wait state.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

---

## AbEnd Field

Returns a Boolean value indicating if the **WebDevice** is to be terminated.

```
BegProp AbEnd Access(*Public) Type(*Boolean)  
    BegGet
```

## Returns

Boolean. **True** if the **WebDevice** is to be abnormally terminated, otherwise **false**.

## Remarks

See the **Detach** method.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [Members](#) | [Detach Method](#) | [ASNA.Monarch Namespace](#)

---

## AbEndMessage Field

Returns the abnormal device terminate message.

```
BegProp AbEndMessage Access(*Public) Type(*String)  
    BegGet
```

## Returns

String. An abnormal termination message.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

## AbEndStack Field

Returns the stack from the terminating application.

```
BegGet AbEndStack Access(*Public) Type(*String)  
BegGet
```

### Returns

String. The stack from the terminating application.

### Remarks

A stack is region of reserved memory in which the applications has stored status data, passed parameters, and sometimes local variables.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

---

## DeviceIsOpen Field

Returns the open or closed status of the **WebDevice**.

```
BegProp DeviceIsOpen Access(*Public) Type(*Boolean)  
BegGet
```

### Returns

**True** if the device is open, otherwise **false**.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDevice Class](#) | [WebDevice Class Members](#) | [ASNA.Monarch Namespace](#)

---

## WebDisplayFile Class

The **WebDisplayFile** class.

For a list of all members of this type, see [WebDisplayFile Members](#).

ASNA.Monarch

### ASNA.Monarch.WebDisplayFile

```
public class WebDisplayFile Inherits System.Object
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA.VisualRPG.Runtime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ASNA.Monarch Namespace](#) | [WebDisplayFile Class Members](#)

---

## WebDisplayFile Method (string, ASNA.DataGate.Client.AdgDataSet)

Initializes a new instance of a **WebDisplayFile** object.

```
BegFunc WebDisplayFile Access(*Public) Type(WebDisplayFile)  
  DclSrParm fileName Type(*String)  
  DclSrParm dataSet Type(ASNA.DataGate.Client.AdgDataSet)  
  DclSrParm shared Type(*Boolean)
```

## Parameters

### fileName

String. The name of the '.aspx' file containing the web form.

### dataSet

**ASNA.DataGate.Client.AdgDataSet**. The dataset associated with the web form.

### shared

Boolean. Specify **True** to allow sharing, otherwise specify **False**.

## Returns

A new instance of a **WebDisplayFile** object for the file and dataset.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

## Close Method

Closes the '.aspx.' web form program and dataSet for the **WebDisplayFile**.

```
BegSr Close Access(*Public) Type(WebDisplayFile)
```

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## GetCurrentRow Method

Returns the current row number for the format name specified.

```
BegSr GetCurrentRow Access(*Public)  
DclSrParm formatName Type(*String)
```

## Parameters

**formatName**

The format name to get the row number for.

## Returns

Integer value containing the row number of the format requested.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## InitMessageSubfile Method(string, string)

**OBSOLETE.** Use **InitMessageSubfile**(string, string, char[])

```
BegFunc InitMessageSubfile Access(*Public)  
DclSrParm subfileName Type(*String)  
DclSrParm ProgramQ Type(*String)
```

## Parameters

**subfileName**

**String.** The message subfile name to be associated with the display file.

**ProgramQ**

**String.** The name of the program queue handling the message subfile.

## Returns

A new instance of a **WebDisplayFile** object for the message subfile and program queue specified.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## InitMessageSubfile Method(string, string, char[ ])

Initializes a new instance of a **WebDisplayFile** object for the message subfile and program queue, with option indicators specified.

```
BegFunc InitMessageSubfile Access(*Public)
    DclSrParm subfileName Type(*String)
    DclSrParm ProgramQ Type(*String)
    DclSrParm optionsIndicators Type(*Char) Rank(1)
```

## Parameters

### subfileName

**String.** The message subfile name to be associated with the display file.

### ProgramQ

**String.** The name of the program queue handling the message subfile.

### optionIndicators

A character array containing the option indicators.

## Returns

A new instance of a **WebDisplayFile** object for the message subfile and program queue, with option indicators specified.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Members](#) | [ASNA.Monarch Namespace](#)

---

## Open Method

Opens the '.aspx.' web form program and dataSet for the **WebDisplayFile**.

```
BegSr Open Access(*Public) Type(WebDisplayFile)
```

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

## Read Method

Reads an array of characters in the **WebDisplayFile** object by the specified indicators.

```
BegSr Read Access(*Public) Type(WebDisplayFile)
DclSrParm indicators Type(*Char) Rank(1)
```

### Parameters

#### indicators

The indicators specified.

### Returns

An array of characters in the WebDisplayFile object containing the values specified, otherwise a null value if the indicators do not exist.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## Update Method

Updates a format record to the **WebDisplayFile** message subfileTable.

```
BegSr Update Access(*Public) Type(WebDisplayFile)
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## Write Method

Writes a format record to the WebDisplayFile message subfileTable.

```
BegFunc Write Access(*Public) Type(WebDisplayFile)
```

### Remarks

One subfile table record for each 'write'.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## AttnID Property

**WebDisplayFile** sets the unique identification for each format record in the message subfile table.

```
BegProp AttnID Access(*Public) Type(*Integer)  
BegSet
```

## Property Value

**Integer.** The unique identification for each format record in the message subfile.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## FeedbackAID Property

Sets or returns the Aid dialog used to provide feedback for a specific field on the webform.

```
BegProp FeedbackAID Access(*Public) Type(*Byte)  
BegGet BegSet
```

## Property Value

**Byte.** The Aid dialog for feedback.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## FeedbackCursor Property

This property returns or sets the cursor coordinates used to provide feedback for a specific field on the webform.

```
BegProp FeedbackCursor Access(*Public) Type(*Short)  
BegGet BegSet
```

## Property Value

**Short.** The cursor coordinates.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## FeedbackField Property

Sets or returns the specific field for which the feedback is to be provided.

```
BegProp FeedbackField Access(*Public) Type(*String)
    BegGet    BegSet
```

### PropertyValue

**String.** The field for which the feedback is to be provided.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## FeedbackFlags Property

Gets the flags used to provide feedback for specific field on the web form.

```
BegProp FeedbackFlags Access(*Public) Type(*Short)
    BegGet
```

### Property Values

**Short.** This property gets the flags used to provide feedback for specific field on the webform.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## ResultingIndicators Property

Sets the indicators used to pass control to the message handler for subsequent action as a result of the feedback.

```
BegProp ResultingIndicators Access(*Public) Type(*String)
    BegSet
```

### Property Value

**String.** The indicators used to pass control to the message handler for subsequent action as a result of the feedback.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [ASNA.Monarch Namespace](#)

---

## Device Field

This field returns the `WebDevice` for the **WebDisplayFile** program.

```
BegSr Device Access(*Public) Type(ASNA.Monarch.WebDevice)
```

## Returns

ASNA.Monarch.WebDevice.

## Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebDisplayFile Class](#) | [WebDisplayFile Class Members](#) | [WebDevice Class](#) | [ASNA.Monarch Namespace](#)

This page intentionally left blank.

## WebJob Class

The **WebJob** class extends the `ASNA.Monarch.Job` class for web applications and provides further controls of program execution and device controls. As part of the session startup, a new job is created, a thread is allocated and the Job is started in this thread. One of the first tasks of job initiation is the creation of a Monarch device instance which is stored as part of the ASP.Net session collections.

For a list of all members of this type, see [WebJob Class Members](#).

ASNA.Monarch

### ASNA.Monarch.WebJob

```
public class WebJob Inherits ASNA.Monarch.Job
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

**WebJob inherits ASNA.Monarch.Job.**

The **WebJob** constructor creates a new instance of `WebJob`.

The **Start** method creates a new instance of an `ASNA.Monarch.WebDevice` object for the job specified.

The **ExecuteStartupProgram** method executes the startup command processing program.

The **EndProgram** method terminates the current `WebJob` program.

The **ShutDown** method terminates the `WebJob` program when an abnormal condition occurs.

The **CurrentWebJob** property returns the current job running for the `WebJob`.

The **Device** field is the name of the `ASNA.Monarch.WebDevice` used by the `WebJob`.

You can use the **ShutDown** method from to shutdown the job. This method ends all active programs, closes the database connections for disk and printer files and then aborts the thread assigned to the job.

## Requirements

**Namespace:** `ASNA.Monarch`

**Assembly:** `ASNA.VisualRPG.Runtime.dll`

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[ASNA.Monarch Namespace](#) | [WebJob Class Members](#)

## WebJob Method ()

Initializes a new instance of a **WebJob** object.

```
BegFunc WebJob Access(*Public) Type(WebJob)
```

## Returns

A new instance of a **WebJob** object.

## Requirements

**Namespace:** `ASNA.Monarch`

**Assembly:** `ASNA VisualRPG.RunTime.dll`

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## See Also

[WebJob Class](#) | [WebJob Class Members](#) | [ASNA.Monarch Namespace](#)

## EndPrograms Method

This method terminates the **WebJob** program.

```
BegSr EndPrograms Access(*Public) Type(WebJob)
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

WebJob Class | WebJob Class Members | ASNA.Monarch Namespace

## ExecuteStartupProgram Method

```
BegSr ExecuteStartupProgram Access(*Protected) Modifier(*Overrides)
```

### Remarks

This method connects to your database, calls the program and disconnects from the database.

### Examples

```
[Visual RPG]
BegClass MyJob Extends(ASNA.Monarch.WebJob) Access(*Public)
  DeclDB Name(MyDataBase) DBName(" *Public/Prod") Access(*Public)
  BegFunc getDataBase Type(ASNA.VisualRpt.Runtime.Database)      +
                                     Access(*Protected)          +
                                     Modifier(*Overrides)
    LeaveSr
  EndFunc

  BegSr ExecuteStartupProgram Access(*Protected) Modifier(*Overrides)
    Connect      MyDataBase
    Call         Custingc
    Disconnect   MyDataBase
  EndSr
EndClass
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

WebJob Class | WebJob Class Members | ASNA.Monarch Namespace

## PopInvocation Method

Returns the last instance of an ASNA.Monarch.Invocation object in the WebJob stack.

```
BegFunc PopInvocation() Access(*Public) Type(ASNA.Monarch.Invocation)
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

---

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

WebJob Class | WebJob Members | ASNA.Monarch Namespace

---

## PushInvocation Method

Creates an instance of an ASNA.Monarch.Program in the WebJob stack.

```
BegFunc PushInvocation Access(*Public) Type(void)
    DeclSrParm program Type(ASNA.Monarch.Program)
```

### Parameters

**program**

ASNA.Monarch.Program.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

WebJob Class | WebJob Members | ASNA.Monarch Namespace

---

## ShutDown Method

This method ends all active programs, closes the database connections for disk and printer files and then aborts the thread assigned to the job.

```
BegSr ShutDown Access(*Public) Type(WebJob)
```

### Remarks

This method ends all active programs, closes the database connections for disk and printer files and then aborts the thread assigned to the job.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

WebJob Class | WebJob Class Members | ASNA.Monarch Namespace

---

## Start Method

This method starts the job name specified on the ASNA.Monarch.WebDevice.

```
BegSr Start Access(*Public) Type(ASNA.Monarch.WebDevice)
```

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebJob Class](#) | [WebJob Class Members](#) | [ASNA.Monarch Namespace](#)

---

### CurrentWebJob Property

This property contains the name of the current running job for the **WebJob** object.

```
BegSr CurrentWebJob Access(*Public) Type(WebJob)
```

### Returns

The name of the current running job for the **WebJob** object.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebJob Class](#) | [WebJob Class Members](#) | [ASNA.Monarch Namespace](#)

---

### Device Field

This field contains the name of the **ASNA.Monarch.WebDevice** used by the **WebJob**.

```
BegSr Device Access(*Public) Type(ASNA.Monarch.WebDevice)
```

### Returns

The name of the **ASNA.Monarch.WebDevice** used by the **WebJob**.

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

[WebJob Class](#) | [WebJob Class Members](#) | [WebDevice Class](#) | [ASNA.Monarch Namespace](#)

## Enumerations

### MessageTypes Enumeration

The **MessageTypes** enumerated constant defines values for the **SendExternalMessage** method and two of the overloaded **SendProgramMessage** methods in the **Program** class.

```
BegEnum MessageTypes Access(*Public)
```

#### Remarks

**MessageTypes** denote values which classify the message.

#### Members

Member	Description	Value
Informational	This is an informational message.	0
Inquiry	This is an inquiry message.	1
Request	This is a request message.	2
Completion	This is a completion message.	3
Diagnostic	This is a diagnostic message.	4
Notify	This is a notification message.	5
Escape	This is an escape message.	6
Status	This is a status message.	7

#### Requirements

**Namespace:** ASNA.Monarch

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

**Assembly:** ASNA VisualRPG.RunTime.dll

#### See Also

Program Class | ASNA.Monarch Namespace

## OverrideOptions Enumeration

The **OverrideOptions** enumerated constant defines file override options within Monarch. Used by the `OverrideFile` and `Job.OverrideFile` methods.

```
BegEnum OverrideOptions Access(*Public)
```

### Remarks

The **OverrideOptions** enumerated constants defines file override options within Monarch. For each option value within the description below, the value type (of the new option value) and the valid type of files are also noted.

### Members

Member	Value Type/ Description / Applies To	Value
ToFile	System.String. The new file object. In the case of WorkStnFile, this refers to the ASP.Net page path name. Applies to DBFile, WorkStnFile, PrintFile.	0
Mbr	System.String. The name of the physical file member within the fileName. Applies to DBFile.	1
WaitRcd	System.Int32. Time, in seconds, specifying the waiting period for a record. Applies to DBFile.	2
Collate	Boolean. True/False. Indicates the output produced by capable printers should be collated when this property is set to True. Applicable only when printing multiple copies, or when the Copies is >1. Applies to PrintFile.	3
Copies	System.Int32. The number of copies to print. See Also Collate. Applies to PrintFile.	4
Duplex	ASNA.DataGate.Common.PrintDuplex - Enumerated constant defining values on duplex (double-sided) printing options. Applies to PrintFile.	5
FormName	System.String. The name of the print format to use, such as "A4", "Letter", "Tabloid", etc. See PaperSize for available print form names. Applies to PrintFile.	6
Orientation	ASNA.DataGate.Common.PaperOrientation - Enumerated constant defining the landscape or portrait orientation of paper in the printer. Applies to PrintFile.	7
PaperLength	System.Int32. The paper length based upon the Orientation of the paper in the printer. These values are in tenths of a millimeter. Applies to PrintFile.	8
PaperSize	ASNA.DataGate.Common.PaperSize - Enumerated constant defining the size of paper. Applies to PrintFile	9
PaperSource	ASNA.DataGate.Common.PaperSource - Enumerated constant defining values on the printer tray that will be used. Applies to PrintFile.	10

PaperWidth	System.Int32. The paper width based upon the Orientation of the paper in the printer. Values are in tenths of a millimeter. Applies to PrintFile.	11
Printer	System.String. The complete path and name of the printer. For example, if the printer is on a network, you must specify the network, then the printer name. For example, "\\Network\HP LaserJet 5M". Applies to PrintFile.	12
Quality	ASNA.DataGate.Common.PrintQuality - Enumerated constant defining values for the quality rating for output produced by the printer. Applies to PrintFile.	13
ReportName	System.String. The name of the report as seen on the printer spool. Applies to PrintFile.	14
Scale	System.Int32. The percentage factor by which the printed output is to be scaled. The printer must support the output to be scaled or resized. Applies to PrintFile.	15
ShareOpenDataPath	Boolean. True/False. Indicator if the display file (active server page) used by this WorkStnFile allows any other to refer to the same display file. Applies to WorkStnFile.	16

### Requirements

**Namespace:** ASNA.Monarch

**Assembly:** ASNA VisualRPG.RunTime.dll

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

## OverrideScope Enumeration

The `OverrideScope` enumerated constant defines values for the level of override. Used by `Job.OverrideFile`, `Job.DeleteOverride`, `OverrideFile`, and `DeleteOverride` methods.

```
BegEnum OverrideScope Access(*Public)
```

### Remarks

`OverrideScope` denote values which classify the level of override.

### Members

Member	Description	Value
Job	Job level.	0
CallLvl	Call level.	1

### Requirements

**Namespace:** ASNA.Monarch.

**Assembly:** ASNA VisualRPG.RunTime.dll.

**Platforms:** Windows 2000, Windows XP Professional, Windows Server 2003 family.

### See Also

ASNA.Monarch Namespace

---

## Chapter 3 - Web Server Controls

---

### DdsCharField Class

**DdsCharField** is a derived class that further defines a character field. It displays as a text box for user input.

For a list of all members of this type, see [DdsCharField Members](#).

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsCharField**

```
public class DdsCharField: Inherits DdsInputField
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **DdsCharField** represents a user-input text box. The **DdsCharField** control has a **CssClass** called **DdsCharField**.

Use the **Length** property to define the length of the character field.

Use the **Value** property to define a field value to the character field at runtime.

---

### DdsDecField Class

**DdsDecField** is a derived class that further defines a decimal field.

For a list of all members of this type, see [DdsDecField Members](#).

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsDecField**

```
Public class DdsDecField Inherits DdsInputField
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

### Remarks

An instance of **DdsDecField** represents a decimal field.

---

### DdsFile Class

The **DdsFile** class is the container object responsible for enabling and processing function keys. It decides which function keys to draw on the form. When the user selects a function key, the appropriate indicators are turned on.

The **DdsFile** control is displayed as a key banner. If a function key is not enabled, the function keys in the banner are grayed out.

For a list of all members of this type, see [DdsFile Members](#).

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsFile**

```
public class DdsFile Inherits System.Web.UI.Control
```

### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

**DdsFile** has two important properties **AttnKeys** and **FuncKeys**

The **AttnKeys** property controls which keys the user can press, the message to be displayed, and the response indicators to turn on. The attention key settings are in a semi-colon(;) separated list of function key, 'text', and response: option indicators.

- For example: `F10 'Finished' 07 : !99;F4 'Skip' *NONE : 09;`
- This property is mainly used to control processing within the application to control flow within the form since NO input data is transmitted from the browser.

The **FuncKeys** property contains all function key settings in a semi-colon(;) separated list of function key, 'text' and response: option indicators.

- For example: `F3 'Add' 07 : !99;F4 'Delete' *NONE : 09;`
- This property is mainly used to control processing between applications since input data is transmitted from the browser.

---

## DdsInputField Class

**DdsInputField** is a derived class that further defines properties and methods common to display file fields. In addition, **DdsInputField** is also a base class for **DdsCharField**, **DdsDecField** and **DdsTimeStamp**.

For a list of all members of this type, see **DdsInputField Members**.

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsField**

**ASNA.Monarch.WebDspF.DdsInputField**

```
Public class DdsInputField : Inherits ASNA.Monarch.WebDspF.DdsField
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

In addition to the methods, properties and fields of **ASNA.Monarch.WDspFile.DdsField**, there are additional methods and properties specific to input fields.

---

## DdsLink Class

The **DdsLink** class defines a hyperlink field.

For a list of all members of this type, see **DdsLink Members**.

ASNA.Monarch.WebDspF

**ASNA.Monarch.WebDspF.DdsLink**

```
Public class DdsLink Inherits System.Web.UI.WebControls.WebControl
```

## Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

## Remarks

An instance of **DdsLink** represents a hyperlink field.

---

## DdsRecord Class

The **DdsRecord** base class that is a container for the record fields and constants. The record is displayed as a panel with a grid layout.

For a list of all members of this type, see [DdsRecord Members](#).

ASNA.Monarch.WebDspF

### **ASNA.Monarch.WebDspF.DdsRecord** Derived classes

```
Public class DdsRecord: Inherits System.Web.UI.WebControls.Panel
```

#### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

#### Remarks

An instance of **DdsRecord** represents a container for the fields and constants of the record. This class also inherits the methods and properties of **System.Web.UI.WebControls.Panel**.

---

### DdsSubfile Class

**DdsSubfile** is a derived class that represents a Subfile control.

For a list of all members of this type, see [DdsSubfile Members](#).

ASNA.Monarch.WebDspF

### **ASNA.Monarch.WebDspF.DdsSubfile**

```
public class DdsSubfile Inherits DdsRecord
```

#### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

#### Remarks

An instance of **DdsSubfile** represents a Subfile control.

---

### DdsSubfileControl Class

The **DdsSubfileControl** class is an extension of a **DdsRecord** for the Subfile. It is both the controller and container of a subfile.

For a list of all members of this type, see [DdsSubfileControl Members](#).

ASNA.Monarch.WebDspF

### **ASNA.Monarch.WebDspF.DdsSubfileControl**

```
public class DdsSubfileControl Inherits DdsRecord
```

#### Thread Safety

Any public static (Shared) members of this type are safe for multithreaded operations. Any instance members are not guaranteed to be thread safe.

#### Remarks

An instance of **DdsSubfileControl** represents an extension of a

This page intentionally left blank.

## Chapter 4 - Operation Codes

### New Op Codes

#### DCLWORKSTNFILE

Declares an externally described workstation file for processing in a program.

```
DCLWORKSTNFILE
Name (name)
DspFile (*NAME | Character Expression)
ImpOpen(*YES | *NO)
RnmFmt (*NONE | Old Format name, New Format name, ...)
Subfile(*NONE | Format name, RRN name, ...)
Prefix (prefix name, [ Length ])
Access (*PRIVATE | *PUBLIC | *PROTECTED | *INTERNAL)
Shared (*NO | *YES)
ShareOpen (*NO | *YES)
```

#### Parts

##### Name

Required. **Name** of the workstation file.

##### DspFile

Optional. For ASP.Net based display files, the **DspFile** keyword names the '.aspx' file containing the web form. The character expression may contain an absolute or relative path to where the file is located; for relative paths, the location of the program containing the DclWorkStnFile is considered the 'current' folder.

##### ImpOpen

Optional. **ImpOpen** specifies if the file will be implicitly opened. The default is \*YES.

- \*YES (default) - specifies that the file will automatically be opened when the program is first run. If the implicit open is unsuccessful, a run-time error will occur.
- \*NO - specifies that the file will not automatically be opened when the program is first run. The RPG IV USROPN keyword is supported in Visual RPG by using ImpOpen(\*NO).

##### RnmFmt

Optional. **RnmFmt** renames the external formats to new, internal formats to avoid clashes with duplicate format names. \*NONE is the default.

- For multi-format files, you enter the external and internal format names in pairs. All names are separated by commas.
- For single format files, you can either specify the internal format name as the only format name, or enter both the external and internal format names separated by a comma.

```
RnmFmt ( RCMMaster, CustInfo)           // single format
RnmFmt ( CustInfo)                     // single format
RnmFmt ( RSubCust, CustSub, RSubSales, SalesSub) // multi-format
```

##### Subfile

Optional. The **SubFile** keyword is used to internally define the subfiles that are specified in the workstation file. The Format Name identifies the name of the subfile record format and the RRN Name identifies the name of the numeric field to be associated with the subfile to communicate the relative-record-number. Each subfile format name, with its corresponding RRN field name, must be listed in the single SubFile keyword.

The relative-record number of any record retrieved by a READC or CHAIN operation is placed into the field identified by the RRN parameter. This field is also used to specify the record number used in WRITE operations to the subfile. The field name specified as the RRN field parameter must be defined as numeric with zero decimal positions.

### Prefix

Optional. **Prefix** allows the specification of a string which is to be prefixed to the field names of an externally described record format. The characters specified as (prefix name) are prefixed of all fields defined in all record formats of the file.

You can also optionally specify a numeric value (length) to indicate the number of leading characters in the existing name to be replaced. If a length is not specified or is zero, the string is prepended to the name.

### Access

Optional. The type of **access** to the workstation file. \*PRIVATE is the default.

- \*PRIVATE (default) - access to the workstation file is accessible only from within their declaration context, including from members of any nested types, for example from within a nested procedure or from an assignment expression in a nested enumeration.
- \*PUBLIC - access to the workstation file is accessible from anywhere within the same project, from other projects that reference the project, and from an assembly built from the project.
- \*PROTECTED - access to the workstation file is accessible only from within the same class, or from a class derived from this class.
- \*INTERNAL - access to the workstation file is accessible from within the same project, but not from outside the project.

### Shared

Optional. **Shared** specifies whether or not this file is to be shared. \*NO is the default.

- \*NO (default) - this workstation file will not be shared.
- \*YES - this workstation file will be shared.

### ShareOpen

Optional. **ShareOpen** specifies if the display file (active server page) used by this DclWorkStnFile allows any other to refer to the same display file. Valid values are \*Yes and \*No. \*No is the default.

- \*NO (default) - this display file can not be shared.
- \*YES - When specifying yes, two or more programs can share the same instance of the display file, that is, one program can write records to the file, and the other can read the data provided by the user.

### Remarks

This command declares an externally described workstation file for processing in a program.

```
DclWorkStnFile CUSTDSPF DspFile("./CUSTDSPF.aspx") Subfile( CustSfl, CustRrn,  
OrderSfl, OrderRRN )
```

### See Also

Web Server Controls | IMPDSPFILE

---

## EXFMT

The specified form is made the "active form". The form is displayed modally and the form is made the "active form". The program executing this statement will wait until the **EXFMT** has been completed.

If the form program is not already loaded, **EXFMT** creates the form, executes its mainline code, and makes it visible to the user. If the form program is already created, **EXFMT** makes it visible to the user.

```
EXFMT  
Fmt (Form Name)  
Err (Indicator Variable | *EXTENDED)
```

### Parts

#### Fmt

Required. Contains the name of the form to display.

## Err

Optional. Determines what the runtime will do if an error occurs while executing this command. The following rules apply:

- **Err** (Indicator Variable) - The specified Indicator is set if an error occurred.
- **Err** (\*EXTENDED) - A program status flag is set whose status can be checked using the **%ERROR** built-in function.
- If the **Err** keyword is not given and an error occurs, a runtime exception will be thrown, terminating the program.

## Remarks

**EXFMT** first creates the form and **SHOWs** the form, and locks out all other input to any other forms in the application.

**EXFMT** behaves just like **SHOW**, however the code following the **EXFMT** statement will not be executed until the form has been closed.

When the form is closed (by the user or by the programmer), the form in the application issuing the **EXFMT** is unlocked, and the code following the **EXFMT** is then executed.

## Example

```
/* All Exfmt operations present a MODAL form, and locks out all other input to any
other forms in the application. */
DoWhile *In03 = *Off
/* Exfmt shows the prompt screen and expects the user to enter an option. The form,
frmListName, is the form name defined in the project and SORTOPT is a field defined
in the form. */
Exfmt frmListName
  Select
    When Option = Add
      Exsr AddRecord
    When Option = Delete
      Exsr DeleteRecord
  Ends1
```

## See Also

RETURN

---

## RETURN

Causes a return to the caller. If used in a form program, it causes a return back to the event cycle.

**RETURN**

## Remarks

If LR is on, the program ends normally, all files are closed, and all data allocated for the program is released back to the system. If LR is off, the program returns to the event handler for that form.

If the program is a form program and LR is off, the program returns to the event cycle. If LR is on, the form unloads, and control returns to the caller if the form was displayed with an **EXFMT**, or a **CALL** command.

## See Also

EXFMT

This page intentionally left blank.

## Improved Operation Codes

### CALL - Call a Program

Transfers program control to an AS/400 program.

```
CALL
Pgm (Character Expression | Form variable)
ParmList (*NONE | Parameter List Name)
Db (*NONE | Database Object)
Err (Indicator Variable | *EXTENDED)
Lr (Indicator Variable)
```

#### Parts

##### Pgm

Required. **Pgm** specifies the name of the program or form that control is transferred to. This can either be an AS/400 program or a local program that is part of the same project.

##### ParmList

Optional. **ParmList** specifies the parameter list name. \*NONE is the default and indicates that there are no parameters to pass.

##### Db

Optional. **Db** specifies the location of the program that is accessed at run-time. \*NONE specifies that the local call is to an ASNA.Monarch.Program or ASNA.Monarch. For a remote call, Db specifies a Database Object defined with the **DCLDB** command.

##### Err

Optional. **Err** determines what the runtime will do if an error occurs while executing this command. The following rules apply:

- **Err** (Indicator Variable) - the specified Indicator is set if an error occurred.
- **Err** (\*EXTENDED) - a program status flag is set whose status can be checked using the %ERROR built-in function.
- If the **Err** keyword is not given and an error occurs, a runtime exception will be thrown, terminating the program.

##### Lr

Optional. The **Lr** parameter will be set when the last record is read.

#### Remarks

The command CALL can be applied to a class name. The class must provide a Shared subroutine called \*Entry which receives its parameters by reference. The parameters mentioned in the subsequent PARM commands will be passed by reference to \*Entry; if it is necessary, any constant parameters mentioned in the PARM statements will be copied to local temporaries to allow the by reference mechanism. The generated subroutine call will be wrapped in a TRY with a CATCH TYPE (ASNA.VisualRPG.Runtime.Return) to allow the called program to issue RETURN commands.

##### Calling an AS/400 Program:

When calling an AS/400 program, PGM should include the library name or \*LIBL, i.e. CALL PGM('Library/Program') or CALL PGM(\*LIBL/Program'). The Call will use the library list associated with the user ID associated with the database name. The library list can be changed by calling the appropriate OS/400 command.

The **maximum number** of parameters that can be passed in an AS/400 call is **36**. The parameters can be of any simple types (packed, char, date, zoned etc...) or data structures or multiple occurrence data structures.

The AS/400 provides for three parameter data types for Commands (e.g., QCMDXC) and Programs (CL, RPG, etc): Character, Boolean and Numeric. Typically, numeric parameters are always Packed. You must explicitly declare numeric parameters as Type(\*packed) in your ASNA Monarch code. It may not be presumed that the ASNA Monarch compiler generates packed numerics.

You can also call AS/400 API's which can require a data structure as a Parm. This is supported and all required data types are permitted. When redefining data structure subfields, only the first definition is processed and converted between ASCII and EBCDIC. See example below:

**For Example:**

```
dclDS Name(MYparmDS)
dcldsfld FieldA Start(1) *char 10          // This value is passed & converted
dcldsfld FieldB Start(1) *packed len(7,2) // Ignored
dcldsfld FieldC Start(5) *zoned len(9,2) // Ignored
```

**Returning from a Form Program:**

A form program returns to the caller when:

- The user closes the form.
- The form is unloaded with the **UNLOAD** op code.
- The form is hidden with the **HIDE** op code(a called form program is automatically UNLOADED with the Hide op code).
- The program returns with the \*INLR indicator on.

**Returning from a Program (VRP) or an AS/400 Program:**

A program returns when:

- The mainline code finishes.
- The **RETURN** op code is executed.

If \*INLR is off when a program returns, all the files are kept open, and all the variables remain intact. If \*INLR is off when returning from a form control, the form remains intact, and control is passed to the event cycle to process user input. Subsequent calls to this program are faster.

**Example**

```
DclFld TimeOfDay *CHAR Len(10)
DclFld DayOfWeek *CHAR Len(10)
DCLFLD MonthOfYear *CHAR Len(10)
// Simple call to a program with 1 Parm being passed.
Call Pgm('MyProgram')
DclParm Name(TimeOfDay)
// Simple call to a program with a Parm List.
DclPlist Listout
Dclparm TimeOfDay
Dclparm DayOfWeek
Dclparm MonthOfYear
Call Pgm('MyProgram') ParmList(Listout)
// Simple call to a form with a Parm List.
DclPlist Listout
Dclparm TimeOfDay
CallMyForm ListOut
// Simple call to AS/400 with a defined constant program name and a complex
parameter list.
DclPlist Listout
Dclparm TimeOfDay
Dclparm DayOfWeek
Dclparm MonthOfYear
DclConst Program1 'Worklib/ZBgTime'
Call Program1 ListOut MyDBName
// Same call as in above example, except a literal constant is used for the called
program.
Call 'Worklib/ZBgTime' ListOut MyDBName
/* Same call as in the above example, except an element is used in an array to
define a called program from within a select group. */
DclArray MyArray DCLFLD(20) Len(20)
Begsr Initialize
    MyArray[1] = 'Worklib/InventoryPgm'
    :
```

```

      :
      MyArray[1] = 'Worklib/ZBgTime'
      :
Endsr

BegSr CallRoutine
  MYDBName = 'S1037242'
  Select
    When X = 5
      Call MyArray[X] ListOut MYDBName
    :
    When X = 10
      Call MyArray[x] ListOut MYDBName
    :
  Ends1
Endsr

```

## PARM - Identify Parameters

Defines the parameters that compose a parameter list that will be passed to a subprogram when a **CALL** command is issued, or will be passed to a subroutine when an **EXSR** command is issued.

**For ASNA Monarch** - This command must be implemented for local calls.

```

PARM
Name (Variable name)
CpyFrom (*NONE | Variable | Literal)
CpyTo (*NONE | Variable)
Type (*NONE | *BINARY | *BOOLEAN | *BYTE | *CHAR | *FLOAT | *FLOAT4 | *FLOAT8 |
*IND | *INTEGER | *INTEGER2 | *INTEGER4 | *INTEGER8 | *ONECHAR | *PACKED | *ZONED)
Len (Length Integer, [Decimal Integer])
DbDirection (*BOTH | *INPUT | *OUTPUT)

```

### Parts

#### Name

Required. **Name** contains the name of the parameter field to pass.

Visual RPG allows you to name a control property as a parameter field, provided the property conforms to valid **PARM** types. However, you must not assume the default property for the control. Instead, you must specifically code the desired property, as shown below.

```

PARM Name ( ioField1.value )
PARM Name ( checkbox1.value )

```

#### CpyFrom

Optional. **CpyFrom** contains the value of the parameter. \*NONE is the default. See Remarks section for more information.

#### CpyTo

Optional. **CpyTo** contains the Parm values are copied from **CpyFrom**. \*NONE is the default.

#### Type

Optional. **Type** can be any of the types listed (Character, Packed or Zoned). \*NONE is the default. See **Type Parameter** for further information.

#### Len

Optional. **Len** defines the length of the field. If **Len** is decimal (3,1) \*ZONED is assigned as the default. If **Len** is character (3), \*CHAR is assigned as the default.

Depending upon the **Type** specified, the **Len** parameter may be required. For instance, Types of \*CHAR, \*PACKED, \*ZONED will require a **Len**. A compiler message will display if the **Len** parameter is needed.

#### DbDirection

Optional. **DbDirection** specifies whether the parameter to a remote program call is an input-only parameter (\*Input), an output-only parameter (\*Output) or an input and output parameter (\*Both). When a large amount

of data is passed in only one direction - either to the called program, or from the called program, **DbDirection** allows you to reduce the call time by eliminating the transmission of unused data. \*Both is the default.

The direction of the parameter is from the perspective of the program being called. That is, \*Input means the called program is expecting data in the parameter, but doesn't need to return it to the caller. \*Output means the called program will send data back in the parameter to the caller, but is not expecting data in the parameter as input to it.

**DbDirection** has specific meaning depending upon the type of the Database to which the Call is being made.

- **Calling AS/400 programs** - AS/400 programs do not specify whether the parameter is used by the program for input or output, so the real value of DbDirection is in time savings when the parameter is large. Specifying the proper direction when data is only used in one direction saves time because the data in the parameter is only sent to the AS/400 when the parameter is marked as \*Input and only transmitted back to the client if the parameter is marked for \*Output.
- **Calling SQL Server stored procedures** - SQL Server defines two kinds of parameters, regular parameters and 'output' parameters. SQL Server regular parameters are received by the procedure so they should have a **DbDirection** of \*Input. What SQL Server means by an 'output' parameter is really a parameter that is both received and sent back to the client, so this kind of parameter should have a **DbDirection** of \*Both. The **DbDirection** must match the stored procedure definition, otherwise an error will be generated.

#### DBDirection Example:

If an SQL Server Stored procedure looks like the following:

```
CREATE PROCEDURE CountNewOrders
@WearhouseId as decimal(8,0) ,
@OrderCount as numeric (5,0) output
AS
select @OrderCount = count(distinct(OrderID)) from [Orders] where WId=@
WearhouseId
GO
```

Then the Visual RPG Call should be:

```
CALL PGM ('CountNewOrders') DB ('SQL')
PARM OrderNum DbDirection (*Input)
PARM OrderCount DbDirection (*BOTH)
```

#### Remarks

The **PARM** command comes after a CALL, DCLPLIST, or EXSR command, and defines the parameters that will be passed to the subprogram/subroutine when it is called. One or more **PARM** commands can come after a CALL or DCLPLIST command.

A **PARM** command that does not come after a CALL, DCLPLIST or another **PARM** command will cause a compiler error.

When a program is called, any parameters specified are assigned the value of the operand specified in **CpyFrom**. The parameter is then passed to the program by reference, meaning that the program may change the value of the Parm. Finally, the Parm values are copied to any **CpyTo** operands specified on the CpyTo keyword.

The data type and length can be specified.

You can pass a data structure as a parameter to an AS/400 program when doing a remote call. There are, however, some restrictions as noted below, on the format of the data structure due to the fact that these fields need to be converted between PC and AS/400 formats (things like ASCII to EBCDIC, byte order, etc.). Note that the following restrictions don't apply for calling programs within the same project.

- All of the space taken by the data structure should be described. If an area of the data structure is not occupied by a field, the AS/400 program will not get the contents of that area. If the AS/400 program were to modify that area, then the PC program will not get the modification.
- Data structure fields cannot be arrays.

#### Example

```
// Call program GetDate passing it a named parameter list.
CALL PGM('GetDate') PARMLIST(IoDateValues)
```

```
// Declare the parameter list for the above call. The value of DayYear is copied to
IoDayYear before the CALL is processed, and the return value is then placed in
field IoYearMonthDay. These fields are declared elsewhere. */
DCLPLIST Name(IoDateValues)
PARM Name (IoMonth)
PARM Name (IoCompleteDate) CPYFrom(DayYear) CPYTO(IoYearMonthDay)
// The following call is identical in function to the above call, but passes the
parameters directly, rather than using a names parameter list. */
CALL PGM('GetDate')
PARM Name (IoMonth)
PARM Name (IoCompleteDate) CPYFrom(DayYear) CPYTO(IoYearMonthDay)
// The following is an example of calling a subroutine and passing parameters.
EXSR ComputeTax
PARM Name (TaxRateCode)
PARM Name (Value)
```

**See Also**

CALL