

# ***Polydoc Web***

## ***Polydoc Web – API (Programmer’s Guide)***

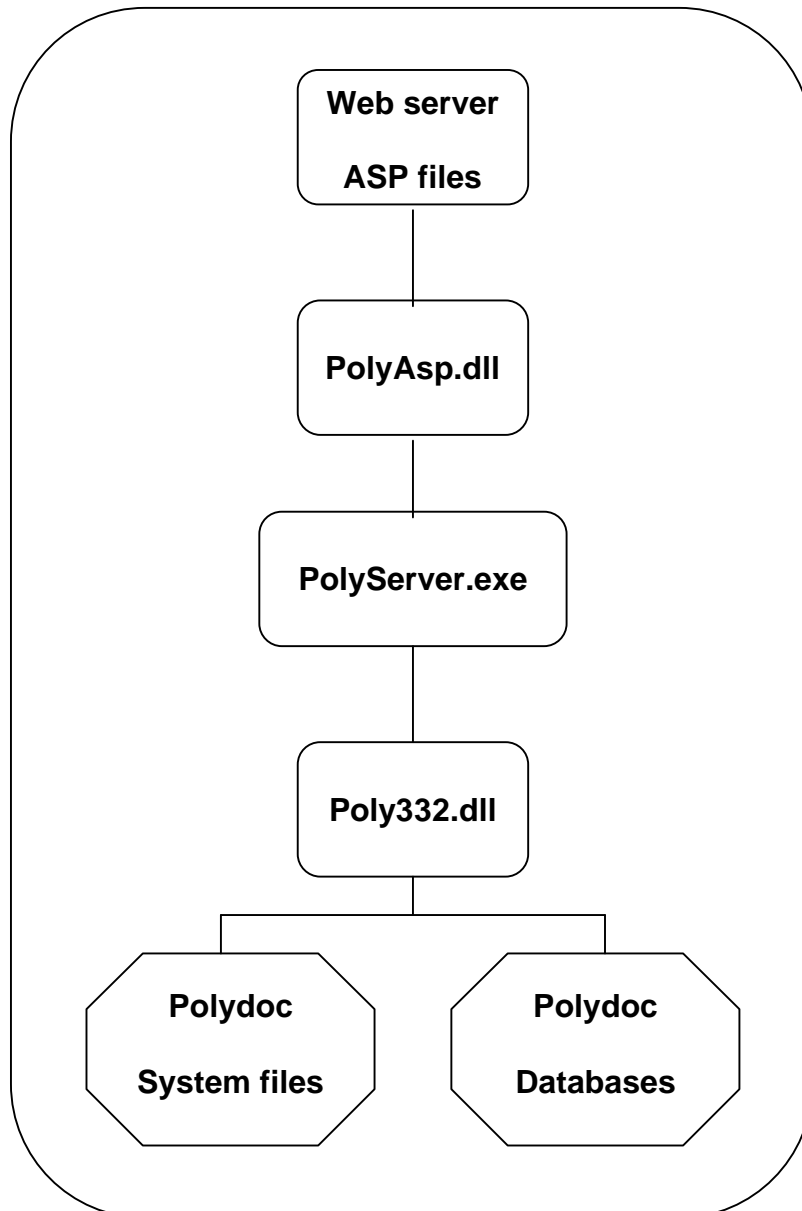
**Contents**

	<b>Page</b>
Introduction .....	3
Starting Polydoc.....	4
User .....	5
Database .....	6
Fields.....	7
Search Terms .....	10
Search .....	11
History (Search Sets) .....	12
Search Result (View Records).....	14
Version Information .....	16
Errors.....	17
Statistics – System Information.....	18
Statistics – Active Users .....	20
Statistics – Users .....	21
Statistics – Databases.....	22
HTML – Presentations.....	23

## Introduction

Polydoc Web consists of a server software called PolyServer.exe, a client software called PolyAsp.dll, and a web interface containing ASP and HTML files. When a session is opened on a Polydoc Web site, the PolyAsp object (working "in process") is loaded by the web server. PolyAsp then handles all calls to PolyServer (working "out of process") through the COM/DCOM concept. PolyServer.exe is then calling Poly332.dll to exchange information with Polydoc system files and database files.

This documentation describes the API (Application Programming Interface) of PolyAsp.dll.



## Starting Polydoc

To use Polydoc on ASP pages, you should first create a handle to a Polydoc object. It is usually done in the file *global.asa*:

```
<script Language="VBScript" runat="server">
sub Session_OnStart      ' Is being called when a client connects
  </script>
  <OBJECT RUNAT=SERVER SCOPE=Session ID=Poly PROGID="PolyAsp.Poly">
  </OBJECT>
  <script Language="VBScript" runat="server">
    Poly.SessionStart
  end sub
sub Session_OnEnd      ' Is being called when a Session.Timeout has occurred,
  ' Poly.SessionStop    ' or when Session.Abandon has been called.
end sub
```

The Polydoc object is available at session level and referred to as "**Poly**" in this documentation.

### **Poly.SessionStart** (Method)

After creating the Polydoc object, you should call `Poly.SessionStart` to connect Polydoc Web with Polydoc and to open a Polydoc session. This method may take two parameters:

`Poly.SessionStart (SystemFolder As String, StatisticsName As String)`

The first parameter is ignored if not running a Webhotel Edition of Polydoc Web. The second parameter can be used to specify a statistics name.

### **Poly.SessionStop** (Method)

To stop current Polydoc session, you may call `Poly.SessionStop`.

Note: It is not necessary to call this function at `Session_OnEnd` in *global.asa* because it is automatically done by the web server when releasing memory used by the session.

## User

The following functions are available for users.

**Poly.Login (Name As String, Password As String)** (Method)

Logs in to Polydoc with entered name and password.

**Boolean = Poly.LoginRequired** (Property)

Returns True if login is required. When the Polydoc system contains a single user account the login is handled automatically. False is also returned when current session is already logged in.

**Boolean = Poly.NeedToLogin** (Property)

This property is included for compatibility with earlier releases.  
Please use Poly.LoginRequired instead.

**Integer = Poly.UserSystemRights** (Property)

Returns the system permission level of current user.

## Database

The following functions are available for databases.

**String = Poly.GetFirstBaseName** (Property)

Returns the name of the first available database.

**String = Poly.GetNextBaseName** (Property)

Returns the name of the next available database. The result is "" when no more databases are available. Example:

```
strBaseName = Poly.GetFirstBaseName
Do While strBaseName <> ""
    ...
    strBaseName = Poly.GetNextBaseName
Loop
```

**Poly.OpenBase (DatabaseName As String)** (Method)

Opens specified database. Poly.Error can be used to verify the result.

## Fields

The following functions are available for current opened database.

**Integer = Poly.NumberOfFields** (Property)

Returns the number of fields in current database.

**Integer = Poly.SetFirstField** (Method)

Positions to the first field available during search. The result is 0 if no error occurred.

**Integer = Poly.SetNextField** (Method)

Positions to the next field available during search. The result is 0 if no error occurred.

```

Err = Poly.SetFirstField
Do While Err = 0
    ....
    Err = Poly.SetNextField
Loop
  
```

**Integer = Poly.FieldNo** (Property)

Returns the field index of current field.

**Poly.FieldNo = Integer** (Property)

Positions to a field. You can step through each field in the database using this property.

**String = Poly.GetFieldCode** (Property)

Returns the code of current field.

**Integer = Poly.GetFieldLength** (Property)

Returns the length of current field. 0 = unlimited.

**String = Poly.GetFieldName** (Property)

Returns the name of current field.

**Integer = Poly.GetFieldType** (Property)

Returns the type of current field.

0 = Text (Alphanumerical), 1 = Range (Interval), 2 = Numerical, 3 = Reference

**String = Poly.GetFieldFullText** (Property)

Returns the fulltext searchcode of current field, or "" if not defined.

**String = Poly.GetFieldSubField** (Property)

Returns the subfield searchcode of current field, or "" if not defined.

**Integer = Poly.FieldSecurity** (Property)

Returns the access level of current field.

**Boolean = Poly.Fields (Attribute As Integer)**

(Property)

Returns the status of selected attribute.

Attributes:

- 1 Field type is Alphanumerical (Text)
- 2 Field type is Range (Interval)
- 3 Field type is Numerical
- 4 Field type is Reference
- 20 Field is searchable
- 21 Field is fulltext searchable
- 22 Field is subfield searchable
- 23 The fulltext searchcode is among the searchcodes being used for default searching
- 24 The subfield searchcode is among the searchcodes being used for default searching
- 27 Field is part of the Level 1 (Print) definition for viewing a record
- 28 Field is part of the Level 2 (Print Trial) definition for viewing a record
- 29 Field is part of the Level 3 (Print Full) definition for viewing a record
- 30 Field can be accessed by current user
- 31 Field is part of the fields available during search

**Boolean = Poly.IsDefFullText**

(Property)

Poly.IsDefFullText is included for compatibility with earlier releases.

Returns True if the fulltext searchcode of current field is among the searchcodes being used for default searching.

**Boolean = Poly.IsDefSubField**

(Property)

Poly.IsDefSubField is included for compatibility with earlier releases.

Returns True if the subfield searchcode of current field is among the searchcodes being used for default searching.

**Boolean = Poly.IsPrint**

(Property)

Poly.IsPrint is included for compatibility with earlier releases.

Returns True if current field is included in the Level 1 (Print) definition for viewing a record.

**Boolean = Poly.IsTrial**

(Property)

Poly.IsTrial is included for compatibility with earlier releases.

Returns True if current field is included in the Level 2 (Print Trial) definition for viewing a record.

**Boolean = Poly.IsFull**

(Property)

Poly.IsFull is included for compatibility with earlier releases.

Returns True if current field is included in the Level 3 (Print Full) definition for viewing a record.

**Boolean = Poly.FieldsNumber (FieldCode As String)**

(Property)

Returns True if the field type of specified field is Numerical or Range.



**Integer = Poly.FindField (FieldText As String, Type As Integer)** (Method)

Returns the field index of specified field.

FieldText: Field code or field name.

Type:

- 1 Locate field in database.
- 2 Locate field among the fields available during search.

**String = Poly.GetFirstFieldCode (ListType As String, Parameters As Integer)** (Method)

Returns the first field code in selected list of fields available during search.

Poly.GetFirstFieldCode is used by the Search Term list.

ListType:

- "0" All fields
- "1" Default fields
- "FC" Any field code.

Parameters:

- 0 Any field
- 8 Searchable fields
- 16 Include both FCS and FCF
- 32 FCF as preferred item (default is FCS)

Parameters can be ored together. FCS is field code subfield, FCF is field code fulltext.

Example: Poly.GetFirstFieldCode("0",24)

## Search Terms

The following functions are available for search terms.

**Poly.FindIntervalSearchTerm (FieldCode As String, Initial As String, \_  
NumOfTerms As Integer, Direction As Integer)** (Method)

Reads a range of search terms from selected field (field code) in selected direction.  
When selecting Goto, an Initial word/character to read from should be specified.  
When selecting Next or Previous, you may also specify an Initial word/character to read from.

Direction:

- 0 Goto
- 1 First
- 2 Last
- 4 Next
- 8 Previous

**Poly.MoveFirstIntervalSearchTerm** (Method)

Positions to the first search term in current range.

**Poly.MoveNextIntervalSearchTerm** (Method)

Positions to the next search term in current range.

**Boolean = Poly.EOFSearchTerm** (Property)

Returns True if no more search terms are available.

Example:

```
Poly.MoveFirstIntervalSearchTerm  
Do Until Poly.EOFSearchTerm  
...  
Poly.MoveNextIntervalSearchTerm  
Loop
```

**String = Poly.SearchTerm** (Property)

Returns current search term.

**Integer = Poly.CountSearchTerm** (Property)

Returns the number of records current search term can be found in.

## Search

The following functions are available for searching.

**Poly.OperatorAnd = String** (Properties)  
**Poly.OperatorOr = String**  
**Poly.OperatorNot = String**  
**Poly.OperatorTo = String**

Sets the operator name. For instance Poly.OperatorAnd="AND"

**String = Poly.OperatorAnd** (Properties)  
**String = Poly.OperatorOr**  
**String = Poly.OperatorNot**  
**String = Poly.OperatorTo**

Returns the operator name.

**Poly.TruncationCharacter = String** (Property)

Sets the truncation character to be used during search.

**String = Poly.TruncationCharacter** (Property)

Returns the truncation character being used during search. Default = ":"

**Poly.WildcardCharacter = String** (Property)

Sets the wildcard character to be used during search.

**String = Poly.WildcardCharacter** (Property)

Returns the wildcard character being used during search. Default = "?"

**Poly.SearchBase (Text As String, Fields As String)** (Method)

Search for Text in Fields (" " = default fields, "\*" = All fields, Fieldcode = Selected field)

## History (Search Sets)

The following functions are available for search sets.

**Poly.HistoryType = Integer** (Property)

Sets the compression type of the history. 0 = Normal, 1 = Expanded, 2 = Compressed  
"Compressed" is the default.

**Integer = Poly.CountQueryList** (Property)

Returns the number of search sets in the history.

**Poly.MoveFirstQueryList** (Method)

Positions to the first search set.

**Poly.MoveLastQueryList** (Method)

Positions to the latest search set.

**Poly.MoveNextQueryList** (Method)

Positions to the next search set.

**Poly.MovePreviousQueryList** (Method)

Positions to the previous search set.

**Boolean = Poly.EOFQueryList** (Property)

Returns True if no more search sets are available.

Example:

```
Poly.MoveFirstQueryList
Do Until Poly.EOFQueryList
    ...
    Poly.MoveNextQueryList
Loop
```

**Poly.QueryErase** (Method)

Restarts the history list of search sets.

**Poly.QueryList = Integer** (Property)

Positions to a search set.

**Integer = Poly.QueryList** (Property)

Returns current set number.

**String = Poly.QueryText** (Property)

Returns the search statement of current search set.

String = Poly.InfoQueryList (Property)

This property is included for compatibility with earlier releases. Please use Poly.QueryText instead.

**String = Poly.QueryHits** (Property)

Returns the number of found records in current search set.

**Poly.QuerySort (SetNumber As Integer, \_  
Field As String, SortOrder As Integer)** (Method)

Sorts selected set. Field can be a field code or a field name.  
SortOrder is 0 = Unsorted, 1 = Ascending, 2 = Descending.

Syntax: Poly.QuerySort (SetNumber,Field[:Subfield][;Field[:Subfield]],SortOrder)

By default, only the first subfield of the field is used. You may combine several fields and subfields as in the following examples.

Examples:

```
Poly.QuerySort( 10, "AU", 1)
Poly.QuerySort( 10, "AU;TI", 1)
Poly.QuerySort( 12, "TI:3", 2)
Poly.QuerySort( -999, "AU;TI:3", 2)
```

The latest set number can be abbreviated as -999.

## Search Result (View Records)

The following functions are available for viewing records in a search set.

**Poly.MoveFirstQueryListDoc** (Method)

Positions to the first record in current search set.

**Poly.MoveLastQueryListDoc** (Method)

Positions to the last record in current search set.

**Poly.MoveNextQueryListDoc** (Method)

Positions to the next record in current search set.

**Poly.MovePreviousQueryListDoc** (Method)

Positions to the previous record in current search set.

**Boolean = Poly.EOFQueryListDoc** (Property)

Returns True if no more records are available.

Example:

```
Poly.MoveFirstQueryListDoc
Do Until Poly.EOFQueryListDoc
    ...
    Poly.MoveNextQueryListDoc
Loop
```

**Long = Poly.CountQueryListDoc** (Property)

Returns number of records in current search set.

**Poly.DocDisplayFormat = Integer** (Property)

Sets current display format. Used by Poly.GetDocQueryList.  
Formats: 0 = Complete, 11 = Short format, 12 = Long format

**Integer = Poly.DocDisplayFormat** (Property)

Returns current display format. Used by Poly.GetDocQueryList.  
Formats: 0 = Complete, 11 = Short format, 12 = Long format

**String = Poly.DocField (FieldNo As Integer)** (Property)

Returns the contents of selected field in current record.

**String = Poly.DocFieldDisplayFormat (FieldNo As Integer)** (Property)

Returns the contents of selected field in current record. The resulting string is partly prepared for HTML presentation.

**String = Poly.GetDocQueryList** (Property)

Returns the contents of current record. The resulting string is partly prepared for HTML presentation. See also Poly.DocDisplayFormat.

**Poly.QueryListDoc = Long**

(Property)

Positions to a specified record.

**Long = Poly.QueryListDoc**

(Property)

Returns current record number.

## Version Information

The following functions are available for version information.

**String = Poly.WebEdition** (Property)

Returns the edition name.

**String = Poly.Version** (Property)

Returns the version number of Poly332.dll, PolyServer.exe and PolyAsp.dll

**String = Poly.VersionName (Type As Integer)** (Property)

Returns the name of each item.

Type = 1 returns Poly332.dll

Type = 2 returns PolyServer.exe

Type = 3 returns PolyASP.dll

**String = Poly.VersionNumber (Type As Integer)** (Property)

Returns the version number.

Type = 1 returns version number of Poly332.dll

Type = 2 returns version number of PolyServer.exe

Type = 3 returns version number of PolyASP.dll

**Long = Poly.VersionRevision (Type As Integer)** (Property)

Returns the revision number of application components. Can be used to verify the revision of a component.

Type = 1 returns revision of Poly332.dll

(Example: 1146 -> 1146)

Type = 2 returns revision of PolyServer.exe

(Example: 1050185 -> 1.05.0185)

Type = 3 returns revision of PolyASP.dll

(Example: 1050085 -> 1.05.0085)



## Errors

The following functions are available for error handling.

**Long = Poly.Error** (Property)

Returns current error number. 0 if no error.

**String = Poly.ErrMsg** (Property)

Returns current error text. "" if no error.

**Poly.ClearError** (Method)

Resets current error status.

## Statistics – System Information

The following functions are available for statistics – system information.

**String = Poly.StatisticsName** (Property)

Returns the name of current statistics. The global (main) statistics has no name and will return "".

**Poly.StatisticsName = String** (Property)

Changes to a different statistics. Used when browsing available statistics.  
Note: The user must have Supervisor rights in Polydoc to use this property.

**Long = Poly.StatisticsCount** (Property)

Returns the number of statistics available. When only the global (main) statistics is available, 1 is returned.

**Poly.StatisticsMoveFirst** (Method)

Positions to the first statistics.

**Poly.StatisticsMoveNext** (Method)

Positions to the next statistics.

**Boolean = Poly.StatisticsEOF** (Property)

Returns True if no more statistics are available.

Example:

```

Poly.StatisticsMoveFirst
Do Until Poly.StatisticsEOF
    ...
    Poly.StatisticsMoveNext
Loop
    
```

**String = Poly.StatisticsStartTime** (Property)

Returns the start time of the statistics.

**String = Poly.StatisticsGlobalStartTime** (Property)

Returns the start time of the global (main) statistics.

**String = Poly.StatisticsServerStartTime** (Property)

Returns the start time of the server program.

**Long = Poly.StatisticsSessionCount** (Property)

Returns the number of Polydoc Web sessions currently handled by the server.

**Long = Poly.StatisticsPeakSessionCount** (Property)

Returns the peak number of Polydoc Web sessions handled by the server.

**Long = Poly.StatisticsUserCount** (Property)

Returns the number of users currently logged in.

**Long = Poly.StatisticsPeakUserCount** (Property)

Returns the peak number of logged in users.

**Long = Poly.StatisticsUserCountSoftware** (Property)

Returns the number of users logged in since the software started.

**Long = Poly.StatisticsUserCountTotal** (Property)

Returns the number of users logged in since the statistics started.

**Poly.StatisticsClearCounters** (Method)

Resets the statistics.

Note: If several statistics are available, be sure to select the right one.

## **Statistics – Active Users**

The following functions are available for statistics – active users.

**Poly.StatisticsMoveFirstUser** (Method)

Positions to the first user.

**Poly.StatisticsMoveNextUser** (Method)

Positions to the next user.

**Boolean = Poly.StatisticsUserEOF** (Property)

Returns True if no more users are available.

Example:

```
Poly.StatisticsMoveFirstUser
Do Until Poly.StatisticsUserEOF
    ...
    Poly.StatisticsMoveNextUser
Loop
```

**String = Poly.StatisticsUserLoginTime** (Property)

Returns the login time of selected user.

**String = Poly.StatisticsUserName** (Property)

Returns the name of selected user.

**String = Poly.StatisticsUserDatabaseName** (Property)

Returns the name of database opened by selected user.

## Statistics – Users

The following functions are available for statistics – stored user information.

**Poly.StatisticsMoveFirstSavedUser** (Method)

Positions to the first user.

**Poly.StatisticsMoveNextSavedUser** (Method)

Positions to the next user.

**Boolean = Poly.StatisticsSavedUserEOF** (Property)

Returns True if no more users are available.

Example:

```
Poly.StatisticsMoveFirstSavedUser
Do Until Poly.StatisticsSavedUserEOF
    ...
    Poly.StatisticsMoveNextSavedUser
Loop
```

**String = Poly.StatisticsSavedUserName** (Property)

Returns the name of selected user.

**Long = Poly.StatisticsSavedUserCount** (Property)

Returns the number of logins selected user has made.

## **Statistics – Databases**

The following functions are available for statistics – stored database information.

**Poly.StatisticsMoveFirstSavedBase** (Method)

Positions to the first database.

**Poly.StatisticsMoveNextSavedBase** (Method)

Positions to the next database.

**Boolean = Poly.StatisticsSavedBaseEOF** (Property)

Returns True if no more databases are available.

Example:

```
Poly.StatisticsMoveFirstSavedBase
Do Until Poly.StatisticsSavedBaseEOF
    ...
    Poly.StatisticsMoveNextSavedBase
Loop
```

**String = Poly.StatisticsSavedBaseName** (Property)

Returns the name of selected database.

**Long = Poly.StatisticsSavedBaseCount** (Property)

Returns the number of times selected database has been opened.

## HTML – Presentations

The following functions are available for HTML – presentations.

**Integer = Poly.CountChoices (Text As String, Delimiter As String)** (Method)

Returns the number of items (choices) in the Text string.  
Example: Count = Poly.CountChoices("All;Default;Field", ";")

**String = Poly.GetChoice (Text As String, Delimiter As String, \_  
Index As Integer)** (Method)

Returns the text of selected index.  
Example: Choice = Poly.GetChoice("All;Default;Field", ";", 2)

**String = Poly.SetChoice (Text As String, Delimiter As String, \_  
Index As Integer, NewValue As String)** (Method)

Returns the updated list of texts.  
Example: List = Poly.SetChoice(strList, ";", 2, "Normal")

**String = Poly.WebCreateFieldList (ListType As String, \_  
SelectedItem As String, Parameters As Integer, Texts As String)** (Method)

Returns an option list with fields available during search of selected listtype.

ListType:

- "0" All fields
- "1" Default fields
- "FC" Any field code

SelectedItem: Field code / name which should be the selected item.

Parameters:

- 0 SelectedItem is field name
- 1 SelectedItem is field code
- 2 Add &nbsp; after field names
- 4 Include texts All fields/Default fields
- 8 Searchable fields only
- 16 Include both FCS and FCF
- 32 FCF as preferred item (default is FCS)

Texts: Text and values of All fields, Default fields, and Fulltext (see parameters).  
Delimiter is tilde (~).

Parameters can be ored together. FCS is field code subfield, FCF is field code fulltext.  
Example: Poly.WebCreateFieldList("0","NR", 27, " (fulltext)")

**String = Poly.WebCreateOptionList (Texts as String, Values As String, \_  
Delimiter As String, SelectedItem As String, Parameters As Integer) (Method)**

Returns an option list containing given texts and values.

Texts: A delimited list of texts.

Values: A delimited list of values.

Delimiter: A string with one or more characters.

SelectedItem: Text or value of the selected item.

Parameters: 0=SelectedItem is text, 1=SelectedItem is value, 2=Add &nbsp; to texts.

Parameters can be ored together.

Example: Poly.WebCreateOptionList("And;Or;Not;To", "AND;OR;NOT;TO", ";", "OR", 3)