



# **Global Search Developers Guide**

**Rev 2.7.5**

**20<sup>th</sup> October 2005**

## **Introduction**

The goal of the IntraNomic Global Search module is to allow you to retrieve information from anywhere within your organization (e.g. a true organization wide search). To achieve this goal, the IntraNomic Global Search module must be able to search within a wide verity of information sources (e.g. Documents on the network, databases etc).

Out of the box, the IntraNomic Global Search module can be extended to search within your existing "Network Shares" (without writing any code). To allow the IntraNomic Global Search module to search within a wide verity of information sources, the IntraNomic Global Search module can be extended by developing Global Search Add-on's.

## Search Criteria

The IntraNomic Global Search module allows search criteria to be entered in the same format as if you were using an Internet search engine such as Google. The Global Search Add-ons that you develop also use the same Internet search criteria format.

The remainder of this section describes the search criteria format.

The IntraNomic search criteria, is similar to entering search criteria for the main Internet search engines. The following are valid search criteria examples:

1. budget and june and 2003
2. june or july
3. "audit procedure"
4. "audit procedure" and manufacturing
5. policy document

To search for a phrase, the phrase must be enclosed within double quotation marks (e.g. "AUDIT PROCEDURE").

The conditional words of AND / OR can be used to refine the search criteria.

If a space is left between two words outside of the double quotation marks (e.g. POLICY DOCUMENT), the search will assume that the AND condition should be used (e.g. POLICY **AND** DOCUMENT).

**Note:** You can combine words, phrases and conditions within the search criteria.

The search will return a match if the entered criteria matches the start of a word or phrase. For example, if you enter the following search criteria:

"balance sheet" and add

"Global Search" would search for all matches of "balance sheet" or "balance sheets" and the words add or adds or adding.

**Note:** The search is **NOT** case sensitive e.g. expense is the same as Expense or EXPENSE.

## Developer Basics

There is no limit to the number of Global Search Add-ons that you can develop and link to IntraNomic.

You can develop a Global Search Add-on in any language that IIS supports or that can be called via IIS.

A Global Search Add-on uses standard XML to obtain the entered search criteria and standard XML to output the results of the search.

IntraNomic will process each Global Search Add-on, without performing an Internet Explorer screen refresh.

A developer can develop the following types of Global Search Add-on's:

- 1) Search a remote Index Server Catalog
- 2) Search within an internal application database

IntraNomic ships with several Global Search Add-on examples in the following folder:

*InetPub\wwwRoot\IntraNomic\Developer Examples\Global Search*

- 1) *index server\server.asp*

This example searches an Index Server Catalog located on a remote server (e.g. you may already have Index Server Catalog's that are indexing parts of your network). This example can also be used as a template for searching via other search engines.

- 2) *database\pubs.asp*

This example searches within the SQL Server "pubs" database. This example searches selected database columns and can be used as a template for searching within your application databases.

## Developer Basics continued

### 3) *database\pubs-advanced.asp*

This example is similar to the *pubs.asp* example but also searches within TEXT and NTEXT columns. These data types are handled differently to the standard SQL Server data types.

### 4) *database\viewtitle.asp*

This example page is called from the Global Search module when a search result from the “*pubs.asp*” and “*pubs-advanced.asp*” add-ons, are selected.

## Global Search Add-on Interface

### Input: Search Criteria XML

A Global Search Add-on receives an XML string containing the entered search criteria.

The following example ASP code (without error checking) creates an instance of the Microsoft XML4 object and loads the passed search criteria XML:

```
dim objXmlDom
Set objXmlDom = Server.CreateObject("MSXML2.FreeThreadedDOMDocument.4.0")
objXmlDom.async = False
objXmlDom.load(Request)
```

Once the above code runs, the XML4 object now contains the entered search criteria. The following code is used to extract the search criteria from the XML:

```
dim lngMaxResults
dim strIndexServerQuery
dim strUnescapedSearchQuery
dim strEscapedSearchQuery
dim strUnescaped_OR_SearchQuery
dim strEscaped_OR_SearchQuery

lngMaxResults = unescape(objXmlDom.getElementsByTagName("RESULTS").item(0).text)

strIndexServerQuery =
unescape(objXmlDom.getElementsByTagName("INDEX_SEARCH").item(0).text)

strUnescapedSearchQuery =
unescape(objXmlDom.getElementsByTagName("SQL_UNESCAPED_SEARCH").item(0).text)

strEscapedSearchQuery =
unescape(objXmlDom.getElementsByTagName("SQL_ESCAPED_SEARCH").item(0).text)

strUnescaped_OR_SearchQuery =
unescape(objXmlDom.getElementsByTagName("SQL_UNESCAPED_OR_SEARCH").item(0).text)

strEscaped_OR_SearchQuery =
unescape(objXmlDom.getElementsByTagName("SQL_ESCAPED_OR_SEARCH").item(0).text)
```

The following table describes each of the following XML search criteria elements:

#### RESULTS:

This element contains the maximum number of search results to return.

**Global Search Add-on Interface continued.****INDEX\_SEARCH:**

This element contains the search criteria already formatted for Microsoft Index Server. This element can also be used to search via other search engines (you may need to manipulate its format to fit your selected search engine).

**SQL\_UNESCAPED\_SEARCH:**

This element contains the search criteria formatted for a SQL Server WHERE clause, with the actual criteria kept in its original format (e.g. not escaped). This element can be used to execute the actual search.

**SQL\_ESCAPED\_SEARCH:**

This element contains the search criteria formatted for a SQL Server WHERE clause, with the actual criteria stored in an escaped format. This element can be used to execute the actual search.

**SQL\_UNESCAPED\_OR\_SEARCH:**

This element contains the search criteria formatted for a SQL Server WHERE clause, with the actual criteria kept in its original format (e.g. not escaped) and any boolean AND conditions converted to boolean OR conditions. This element can be used to improve the efficiency of the search by reducing the number of rows that the actual search condition needs to process (please see the examples within this document for more information).

**SQL\_ESCAPED\_OR\_SEARCH:**

This element contains the search criteria formatted for a SQL Server WHERE clause, with the actual criteria stored in an escaped format and any boolean AND conditions converted to boolean OR conditions. This element can be used to improve the efficiency of the search by reducing the number of rows that the actual search condition needs to process (please see the examples within this document for more information).

**Global Search Add-on Interface continued.**

Output: Search Results XML

Once your Global Search Add-on has processed the search criteria it should return the search results using the following standard XML:

```
<RESULTS>
  <COUNT></COUNT>
  <ROWS>
    <ROW>
      <DISPLAYROW1></DISPLAYROW1>
      <DISPLAYROW2></DISPLAYROW2>
      <EXECUTE TYPE= ' FILE | URL' ></EXECUTE>
      <EXTICON></EXTICON>
      <TOOLTIP></TOOLTIP>
    </ROW>
  </ROWS>
</RESULTS>
```

The following table describes each of these output elements:

**DISPLAYROW1:**

This element is displayed as the first line of the search results (it can contain HTML).

**DISPLAYROW2:**

This element is displayed as the second line of the search results (it can contain HTML).

**EXECUTE:**

This element is used to define what to do when a search result is selected:

**FILE:** The file to display e.g.  
<EXECUTE TYPE='FILE'>\\Server\Share\folder\filename.ext</EXECUTE>

**URL:** The URL to call e.g.  
<EXECUTE TYPE='URL'>http:\\server\directory\default.asp?displayid=1</EXECUTE>

**EXTICON:**

An optional icon can be displayed by passing a file extension. For example you could pass DOC, XLS, PPT etc. but it must be a file extension that is known to IntraNomic.

**Global Search Add-on Interface continued.**

**TOOLTIP:**

This element contains additional information that is displayed as a tool tip when the cursor is left over a search result.

If your Global Search Add-on encounters a problem, you should return the following XML:

```
<ERROR></ERROR>
```

The following table describes the ERROR element:

**ERROR:**

This element should contain the error message you wish to display (no other XML tags should be passed back to IntraNomic) e.g.

```
<ERROR>Index Server is stopped, an Administrator has been informed</ERROR>
```

## Search within Index Server

The Global Search Add-on can search within an existing Index Server Catalog located on a remote server or it can use a different search engine.

The following example demonstrates how to search an Index Server Catalog located on a remote server (the following example contains no error checking for simplicity):

```
dim objIndexServer
dim objIndexParam

Set objIndexServer = Server.CreateObject("IXSSO.Query")
Set objIndexParam = Server.CreateObject("IXSSO.Util")

objIndexServer.Reset

objIndexServer.Columns = "path, filename, size, docauthor, write"
objIndexServer.SortBy = "rank[d]"
objIndexServer.MaxRecords = cLng(lngMaxResults)
objIndexServer.Query = "{prop name=contents} " & cstr(strIndexServerQuery)

objIndexServer.Catalog = "Your Catalog Name In Here"

objIndexParam.AddScopeToQuery objIndexServer, "\", "deep"

dim objIndexResults

Set objIndexResults = objIndexServer.CreateRecordSet("nonsequential")
```

**Note:** The strIndexServerQuery was extracted from the search criteria XML. The strIndexServerQuery variable is already pre-formatted for Microsoft Index Server.

## Internal Application Database Search

When searching within an Application Database, you cannot use standard SQL to search within the many database columns. If you used a standard SQL statement, the boolean AND / OR conditions would only be applied to each column and not to the complete row.

To allow standard SQL to use the boolean AND / OR conditions on the row, you should first concatenate the required database columns into a single column. This single column can then be used by a standard SQL statement.

**Note:** Before developing a Global Search Add-on to search within an Application Database you must first define the list of columns that will be searched.

You can use one of the following methods to generate this single column:

- 1) Create an overnight job that concatenates the required columns of a database into a single column (this can be used for large databases that frequently change).
- 2) Create triggers within the database that run when data is added, changed or remove. The trigger would recreate the single column for the affected row (this can be used for large databases that change infrequently).
- 3) Create a temporary single column within the Global Search Add-on that is removed once the search is completed. The examples that ship within IntraNomic use this technique as this technique can be easily implemented. The examples contain several time saving techniques that reduce the initial number of rows that need to be concatenated into a single column (e.g. by initially only concatenating potential rows, the search will execute faster as it has less rows to search).

## Internal Application Database Search continued.

The remainder of this section describes how to implement a temporary single column for a Global Search Add-on.

### Step 1:

The first step involves identifying which rows need to be concatenated into the single column. The following SQL statement uses the `_OR_` search criteria to identify the possible rows. It is much faster to concatenate 100 rows compared to a database of 100,000 rows (this routine is called only when the search criteria contains the AND Boolean condition):

```
dim arrDBColumns

arrDBColumns = Array("publishers.pub_name", "publishers.city",
                    "publishers.state", "publishers.country",
                    "titles.title", "titles.type", "titles.notes",
                    "authors.au_lname", "authors.au_fname",
                    "authors.address", "authors.city", "authors.state",
                    "authors.zip")

strSQLInitialWhereClause = "WHERE (" &
                          FormatSQLCriteria(strEscaped_OR_SearchQuery,
                          strUnescaped_OR_SearchQuery, arrDBColumns) & ") "
```

**Note:** The `FormatSQLCriteria` function is a “helper function” that is used to build the SQL WHERE clause. The `FormatSQLCriteria` function is described later within this section.

The `strEscaped_OR_SearchQuery` and `strUnescaped_OR_SearchQuery` variables were extracted from the search criteria XML.

### Step 2:

The second step involves creating a SQL statement that applies the entered search criteria to the single column:

```
dim strSQLFinalWhereClause

strSQLFinalWhereClause = "WHERE " & FormatSQLCriteria(strEscapedSearchQuery,
                                                    strUnescapedSearchQuery, "txt_concat")
```

**Note:** The `strEscapedSearchQuery` and `strUnescapedSearchQuery` variables were extracted from the search criteria XML.

### Internal Application Database Search continued.

#### Step 3:

The third step creates a temporary table that is populated with rows from the SQL WHERE clause in Step 1:

```
arrSQL(0) = "CREATE TABLE #devSearch (" & _
           " result_id VARCHAR(6) NOT NULL," & _
           " display_line1 VARCHAR(80) NOT NULL," & _
           " display_line2 VARCHAR(40) NOT NULL," & _
           " result_order VARCHAR(80) NOT NULL," & _
           " txt_concat NTEXT NULL)"

'Insert the initial matching rows into the #devSearch temporary table

arrSQL(1) = " INSERT INTO #devSearch" & _
           " SELECT " & _
           " titles.title_id," & _
           " titles.title," & _
           " 'Publisher: ' + publishers.pub_name," & _
           " titles.title," & _
           " publishers.pub_name + ' ' + publishers.city + ' ' + _
           " publishers.state + ' ' + publishers.country + ' ' + _
           " titles.title + ' ' + titles.type + ' ' + " & _
           " titles.notes + ' ' + authors.au_lname + ' ' + _
           " authors.au_fname + ' ' + authors.address + ' ' + _
           " authors.city + ' ' + authors.state + ' ' + _
           " authors.zip + ' '" & _
           " FROM titleauthor WITH (NOLOCK) INNER JOIN" & _
           " authors WITH (NOLOCK) ON titleauthor.au_id = " & _
           " authors.au_id INNER JOIN" & _
           " titles WITH (NOLOCK) ON titleauthor.title_id = " & _
           " titles.title_id INNER JOIN" & _
           " publishers WITH (NOLOCK) ON titles.pub_id = " & _
           " publishers.pub_id INNER JOIN" & _
           " pub_info WITH (NOLOCK) ON publishers.pub_id = " & _
           " pub_info.pub_id " & _
           strSQLInitialWhereClause & _
           " ORDER BY titles.title"
```

The SQL SELECT clause populates the “result\_id, display\_line1, display\_line2 and result order” columns, which are used to populate the output for the search results. The “txt\_concat” column is created by concatenating all of the columns that will be searched (using standard SQL + statements).

The SQL statements are stored within an array (arrSQL) as it is quicker to execute multiple SQL statements in one call, rather than having to make multiple calls to execute multiple SQL statements e.g. `ado.execute join(arrSQL, “;”)`

### Internal Application Database Search continued.

**Step 4:**

The fourth step involves applying the actual search criteria to the temporary table and returning only the required number of search results:

```
strSQL = "SELECT DISTINCT TOP " & lngMaxResults & _  
        " result_id, display_line1, display_line2, result_order" & _  
        " FROM #devSearch " & _  
        strSQLFinalWhereClause & _  
        " ORDER BY result_order"
```

**Step5:**

The output from step 4 can be converted into XML, which is then passed back to IntraNomic.

**Step 6:**

The temporary table can be deleted:

```
DROP TABLE #devSearch
```

## Internal Application Database Search continued.

The FormatSQLCriteria function is used to convert the passed search criteria into a SQL WHERE clause. Depending on the database columns you can change the FormatSQLCriteria function to use the Escaped or Unescaped search criteria:

```
Private Function FormatSQLCriteria(strSQLServer_Escaped_Pass, _
                                strSQLServer_UNEscaped_Pass, _
                                arrColumnsPass() )
'
' arrColumnsPass can contain one or many database columns
' if only one column then create a 1 dimension array
'
Dim arrSQLColumn()
if isarray(arrColumnsPass) then
    ReDim arrSQLColumn(UBound(arrColumnsPass))
    Dim intLoop
    For intLoop = LBound(arrColumnsPass) To UBound(arrColumnsPass)
        '
        ' Note: If you need to include a database column that
        ' contains escaped HTML you should use
        ' strSQLServer_Escaped_Pass rather than
        ' strSQLServer_UNEscaped_Pass
        '
        arrSQLColumn(intLoop) = "(" & _
            Replace(strSQLServer_UNEscaped_Pass, _
                "**column**", arrColumnsPass(intLoop)) & ")"
    Next
Else
    Redim arrSQLColumn(0)
    arrSQLColumn(0) = "(" & Replace(strSQLServer_UNEscaped_Pass, _
        "**column**", arrColumnsPass) & ")"
End If
FormatSQLCriteria = Join(arrSQLColumn, " OR ")
End Function
```

**Note:** The FormatSQLCriteria function within the code examples that ship with IntraNomic, explain how to mix escaped and non-escaped columns.

The pubs-advanced.asp example demonstrates how to concatenate database columns that have the TEXT or NTEXT data type.

## Security

Security can be defined in the following places:

- 1) When you use the “Extend Global Search” module to define your Global Search Add-on to IntraNomic you also define a security list. This security list controls who can view the search results from your Global Search Add-on. Those users that are not included within the security list will not be shown the Global Search Add-on (the Global Search Add-on will also NOT execute during a Global Search).
- 2) If you are viewing a document from an Index Server search if you click on the search result but you do not have security to view the document you will not be shown the document (e.g. Windows security is still enforced).
- 3) You can obtain the credentials of the person performing the search by using:

```
Request.ServerVariables("LOGON_USER")
```

You can implement security by checking the credentials of this person.

## Developing a Viewer

When a search result is clicked, the search result can either:

- 1) View the document directly (e.g. the results of an Index Server search)
- 2) Call a viewer to display more details on the search result (e.g. a database search)

If you are developing an Index Server search you can write the following search result XML:

```
<EXECUTE TYPE='FILE'>\\Server\Share\folder\filename.ext</EXECUTE>
```

You must output the document path as a UNC path and not a local path

If you are developing a database search you can write the following search result XML:

```
<EXECUTE TYPE='URL'>http://server/folder/view.asp?displayid=Order0024</EXECUTE>
```

The “displayid” should be set to the database ID of the output row (e.g. “Order0024”). Within this example the “view.asp” page can then access the “displayid” by using:

```
Request.QueryString( "displayid" )
```

The “view.asp” page would read the database and display the database information for the “Order0024” key.

## Linking the Global Search Add-on to IntraNomic

You can link your Global Search Add-on to IntraNomic by using the “Extend Global Search” module (available to the IntraNomic Administrators).

Your Global Search Add-on does not have to be stored on the IntraNomic server. The Global Search module will call a Global Search Add-on across the network. The machine where the Global Search Add-on resides must have IIS installed.

By default, IntraNomic executes with no network privileges, therefore you should set the IIS folder that contains the Global Search Add-on to use “Anonymous Access”. This allows the Global Search Add-on to be called across the network. Please contact StyleTech if you require a more secure solution as IntraNomic can be re-configured to use network privileges.

If you are searching a remote Index Server Catalog, the Global Search Add-on must reside on the same server as Index Server.

**Note:** It is recommended that when you are adding the “Directories” to the Index Server Catalog, you add those “Directories” as UNC paths rather than local paths. If the “Directories” are added as UNC paths, the results of the Index Server search will be returned as a UNC path, which can then be executed correctly from the user’s PC. If the “Directories” are added as local paths, you will need to convert these local paths to UNC paths within your Global Search Add-on e.g. c:\data\january to \\server\share\january.

If you need defined security to access an application database, you should create your Global Search Add-on as a COM / .NET object. You can then inform Windows to execute the Global Search Add-on using a defined security clearance. You would still need to develop an ASP / ASP.NET page but this page would be a wrapper that calls your COM / .NET object from IIS.

## Testing your Global Search Add-on

StyleTech recommend that you test your Global Search Add-on by hard-coding the output from the IntraNomic search criteria XML. This allows you to run the Global Search Add-on as a standalone program.

Once you have created a working version, you can replace the hard-coded values with the standard XML search criteria. You can then link your Global Search Add-on to IntraNomic by using the “Extend Global Search” module (available to the IntraNomic Administrators). If you are using the IntraNomic production environment (not recommended when developing) you should set the security list to yourself only, so no other users can view the search results until you are ready to put the Global Search Add-on live.

**Note:** The IntraNomic license allows you to install a Developers version of IntraNomic, which can be used by developers to create IntraNomic Add-ons.