[MS-ADMWS]:

Search Service Administration Web Service Protocol

Copyright © 2017 Microsoft Corporation

Release: December 12, 2017

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft’s delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/4/2008</td>
<td>0.1</td>
<td>New</td>
<td>Initial Availability</td>
</tr>
<tr>
<td>6/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>10/6/2008</td>
<td>1.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>12/12/2008</td>
<td>1.02</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>7/13/2009</td>
<td>1.03</td>
<td>Major</td>
<td>Changes made for template compliance</td>
</tr>
<tr>
<td>8/28/2009</td>
<td>1.04</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>11/6/2009</td>
<td>1.05</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>2/19/2010</td>
<td>2.0</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>3/31/2010</td>
<td>2.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>4/30/2010</td>
<td>2.02</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>6/7/2010</td>
<td>2.03</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>6/29/2010</td>
<td>2.04</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>7/23/2010</td>
<td>2.05</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>9/27/2010</td>
<td>2.05</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/15/2010</td>
<td>2.05</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>12/17/2010</td>
<td>2.06</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>2.06</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/10/2011</td>
<td>2.06</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>2.7</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>4/11/2012</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/12/2012</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/30/2013</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>Date</td>
<td>Revision History</td>
<td>Revision Class</td>
<td>Comments</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/23/2016</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/29/2016</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/17/2016</td>
<td>2.7</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/19/2017</td>
<td>2.8</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>12/12/2017</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction ............................................................................................................. 14
   1.1 Glossary ............................................................................................................ 14
   1.2 References ....................................................................................................... 18
      1.2.1 Normative References .............................................................................. 18
      1.2.2 Informative References ............................................................................. 19
   1.3 Protocol Overview (Synopsis) ............................................................................ 19
   1.4 Relationship to Other Protocols ....................................................................... 20
   1.5 Prerequisites/Preconditions ............................................................................. 20
   1.6 Applicability Statement .................................................................................... 20
   1.7 Versioning and Capability Negotiation ............................................................ 20
   1.8 Vendor-Extensible Fields .................................................................................. 20
   1.9 Standards Assignments ..................................................................................... 20

2 Messages ............................................................................................................... 21
   2.1 Transport ......................................................................................................... 21
   2.2 Common Message Syntax ................................................................................ 21
      2.2.1 Namespaces ............................................................................................... 21
      2.2.2 Messages .................................................................................................. 22
      2.2.3 Elements ................................................................................................... 22
      2.2.4 Complex Types ........................................................................................ 22
         2.2.4.1 ArrayOfString .................................................................................... 22
         2.2.4.2 ContentSourceDynamicPropsInternal ............................................. 22
         2.2.4.3 CrawlRuleInternal ............................................................................. 23
         2.2.4.4 ContentSourceInternal ................................................................... 26
      2.2.5 Simple Types ............................................................................................. 28
      2.2.6 Attributes ................................................................................................. 28
      2.2.7 Groups ....................................................................................................... 28
      2.2.8 Attribute Groups ...................................................................................... 28
      2.2.9 Exception Types ........................................................................................ 28
         2.2.9.1 ConcurrencyException .................................................................... 28
         2.2.9.2 UpdatedConcurrencyException ....................................................... 28
         2.2.9.3 DeletedConcurrencyException ....................................................... 29
         2.2.9.4 ExternalException ............................................................................. 29
         2.2.9.5 COMException .................................................................................. 29
         2.2.9.6 KeyNotFoundException ...................................................................... 30
         2.2.9.7 ArgumentException ............................................................................. 30
         2.2.9.8 InvalidOperationException ................................................................. 30
         2.2.9.9 InvalidOperationException .............................................................. 30

3 Protocol Details .................................................................................................. 32
   3.1 SearchApplicationWebService Soap Server Details ....................................... 32
      3.1.1 Abstract Data Model ............................................................................... 32
         3.1.1.1 Object hierarchy .............................................................................. 32
         3.1.1.2 Crawler Application ........................................................................ 33
         3.1.1.3 Portal Content Project .................................................................... 34
         3.1.1.4 Content Source ................................................................................. 35
         3.1.1.5 Crawl Rule ........................................................................................ 36
         3.1.1.6 Anchor Content Source ................................................................... 38
         3.1.1.7 Crawl Mapping ................................................................................ 38
         3.1.1.8 User Profile Import Project .............................................................. 38
         3.1.1.9 User Profile Content Source ........................................................... 39
         3.1.1.10 User Profile Import Domain ........................................................... 39
      3.1.2 Timers ........................................................................................................... 39
      3.1.3 Initialization .............................................................................................. 39
      3.1.4 Message Processing Events and Sequencing Rules .................................. 40

[MS-ADMWS] - v20171212
Search Service Administration Web Service Protocol
Copyright © 2017 Microsoft Corporation
Release: December 12, 2017
3.1.4.1 ActivateCrawlRule ............................................................... 42
  3.1.4.1.1 Messages ................................................................. 43
    3.1.4.1.1.1 ActivateCrawlRuleSoapIn ........................................ 43
    3.1.4.1.1.2 ActivateCrawlRuleSoapOut ...................................... 43
  3.1.4.1.2 Elements ................................................................ 44
    3.1.4.1.2.1 ActivateCrawlRule .................................................. 44
    3.1.4.1.2.2 ActivateCrawlRuleResponse ...................................... 44
  3.1.4.2 ActivateProfileImport .................................................... 44
    3.1.4.2.1 Messages ................................................................. 45
      3.1.4.2.1.1 ActivateProfileImportSoapIn ................................... 45
      3.1.4.2.1.2 ActivateProfileImportSoapOut ................................ 45
    3.1.4.2.2 Elements ................................................................ 45
      3.1.4.2.2.1 ActivateProfileImport ........................................... 45
      3.1.4.2.2.2 ActivateProfileImportResponse ................................ 46
  3.1.4.3 AddApp ................................................................... 46
  3.1.4.4 AddContentSource ......................................................... 46
    3.1.4.4.1 Messages ................................................................. 47
      3.1.4.4.1.1 AddContentSourceSoapIn ....................................... 47
      3.1.4.4.1.2 AddContentSourceSoapOut ..................................... 47
    3.1.4.4.2 Elements ................................................................ 47
      3.1.4.4.2.1 AddContentSource ............................................... 47
      3.1.4.4.2.2 AddContentSourceResponse ................................... 48
  3.1.4.5 AddCrawlMapping ......................................................... 48
    3.1.4.5.1 Messages ................................................................. 49
      3.1.4.5.1.1 AddCrawlMappingSoapIn ....................................... 49
      3.1.4.5.1.2 AddCrawlMappingSoapOut ..................................... 49
    3.1.4.5.2 Elements ................................................................ 50
      3.1.4.5.2.1 AddCrawlMapping ............................................... 50
      3.1.4.5.2.2 AddCrawlMappingResponse ................................... 50
  3.1.4.6 AddCrawlRule ............................................................... 50
    3.1.4.6.1 Messages ................................................................. 51
      3.1.4.6.1.1 AddCrawlRuleSoapIn ........................................... 51
      3.1.4.6.1.2 AddCrawlRuleSoapOut ......................................... 51
    3.1.4.6.2 Elements ................................................................ 51
      3.1.4.6.2.1 AddCrawlRule ..................................................... 51
      3.1.4.6.2.2 AddCrawlRuleResponse ....................................... 52
  3.1.4.7 AddExtension ............................................................... 52
    3.1.4.7.1 Messages ................................................................. 53
      3.1.4.7.1.1 AddExtensionSoapIn ............................................ 53
      3.1.4.7.1.2 AddExtensionSoapOut ......................................... 53
    3.1.4.7.2 Elements ................................................................ 53
      3.1.4.7.2.1 AddExtension ..................................................... 53
      3.1.4.7.2.2 AddExtensionResponse ....................................... 54
  3.1.4.8 AddSitePath ................................................................. 54
  3.1.4.9 AddSiteRestriction ......................................................... 54
  3.1.4.10 BackupRegistry ........................................................... 54
    3.1.4.10.1 Messages ............................................................... 55
      3.1.4.10.1.1 BackupRegistrySoapIn ....................................... 55
      3.1.4.10.1.2 BackupRegistrySoapOut ..................................... 55
    3.1.4.10.2 Elements ............................................................... 55
      3.1.4.10.2.1 BackupRegistry ................................................ 55
      3.1.4.10.2.2 BackupRegistryResponse ................................... 55
  3.1.4.11 CatalogPauseCrawlRequest ........................................... 56
    3.1.4.11.1 Messages ............................................................... 56
      3.1.4.11.1.1 CatalogPauseCrawlRequestSoapIn ............................ 56
      3.1.4.11.1.2 CatalogPauseCrawlRequestSoapOut .......................... 56
    3.1.4.11.2 Elements ............................................................... 57
      3.1.4.11.2.1 CatalogPauseCrawlRequest ................................. 57
3.1.4.21.2.2 GetContentSourcesResponse ........................................ 71
3.1.4.21.3 Complex Types .......................................................... 72
  3.1.4.21.3.1 ContentSourcesInternal ........................................ 72
  3.1.4.21.3.2 ArrayOfContentSourceInternal ................................ 72
3.1.4.22 GetContentState .......................................................... 72
  3.1.4.22.1 Messages ................................................................ 73
    3.1.4.22.1.1 GetContentStateSoapIn ...................................... 73
    3.1.4.22.1.2 GetContentStateSoapOut .................................. 73
  3.1.4.22.2 Elements ................................................................ 73
    3.1.4.22.2.1 GetContentState ............................................... 73
    3.1.4.22.2.2 GetContentStateResponse .................................. 74
3.1.4.22.3 Complex Types .......................................................... 74
  3.1.4.22.3.1 ContentStateInternal ........................................... 74
3.1.4.23 GetCrawlMappings .......................................................... 75
  3.1.4.23.1 Messages ................................................................ 75
    3.1.4.23.1.1 GetCrawlMappingsSoapIn .................................... 75
    3.1.4.23.1.2 GetCrawlMappingsSoapOut .................................. 75
  3.1.4.23.2 Elements ................................................................ 75
    3.1.4.23.2.1 GetCrawlMappings .............................................. 75
    3.1.4.23.2.2 GetCrawlMappingsResponse .................................. 76
  3.1.4.23.3 Complex Types ........................................................ 76
    3.1.4.23.3.1 ArrayOfCrawlMappingInternal .............................. 76
    3.1.4.23.3.2 CrawlMappingInternal ....................................... 76
3.1.4.24 GetCrawlRuleList ......................................................... 77
  3.1.4.24.1 Messages ................................................................ 77
    3.1.4.24.1.1 GetCrawlRuleListSoapIn .................................... 77
    3.1.4.24.1.2 GetCrawlRuleListSoapOut .................................. 77
  3.1.4.24.2 Elements ................................................................ 77
    3.1.4.24.2.1 GetCrawlRuleList .............................................. 77
    3.1.4.24.2.2 GetCrawlRuleListResponse .................................. 78
  3.1.4.24.3 Complex Types ........................................................ 78
    3.1.4.24.3.1 ArrayOfCrawlRuleInternal .................................. 78
3.1.4.25 GetExtensionList ........................................................... 78
  3.1.4.25.1 Messages ................................................................ 79
    3.1.4.25.1.1 GetExtensionListSoapIn .................................... 79
    3.1.4.25.1.2 GetExtensionListSoapOut .................................. 79
  3.1.4.25.2 Elements ................................................................ 79
    3.1.4.25.2.1 GetExtensionList .............................................. 79
    3.1.4.25.2.2 GetExtensionListResponse .................................. 80
3.1.4.26 GetGathererApplications ................................................ 80
3.1.4.27 GetImportAccounts ....................................................... 80
  3.1.4.27.1 Messages ................................................................ 80
    3.1.4.27.1.1 GetImportAccountsSoapIn .................................. 80
    3.1.4.27.1.2 GetImportAccountsSoapOut ............................... 81
  3.1.4.27.2 Elements ................................................................ 81
    3.1.4.27.2.1 GetImportAccounts ............................................ 81
    3.1.4.27.2.2 GetImportAccountsResponse ............................. 81
  3.1.4.27.3 Complex Types ........................................................ 81
    3.1.4.27.3.1 ArrayOfImportDomainAccount ............................ 82
    3.1.4.27.3.2 ImportDomainAccount ....................................... 82
3.1.4.28 GetImportStatus ............................................................ 82
  3.1.4.28.1 Messages ................................................................ 82
    3.1.4.28.1.1 GetImportStatusSoapIn ..................................... 82
    3.1.4.28.1.2 GetImportStatusSoapOut .................................. 83
  3.1.4.28.2 Elements ................................................................ 83
    3.1.4.28.2.1 GetImportStatus ............................................... 83
    3.1.4.28.2.2 GetImportStatusResponse .................................. 83
  3.1.4.28.3 Complex Types ........................................................ 84
3.1.4.54  ResumeCrawl .................................................................108
  3.1.4.54.1  Messages ...............................................................109
    3.1.4.54.1.1  ResumeCrawlSoapIn ........................................109
    3.1.4.54.1.2  ResumeCrawlSoapOut .......................................109
  3.1.4.54.2  Elements .............................................................109
    3.1.4.54.2.1  ResumeCrawl .................................................109
    3.1.4.54.2.2  ResumeCrawlResponse ......................................109
3.1.4.55  SearchDatabaseCleanup ................................................110
  3.1.4.55.1  Messages ...............................................................110
    3.1.4.55.1.1  SearchDatabaseCleanupSoapIn ................................110
    3.1.4.55.1.2  SearchDatabaseCleanupSoapOut ...........................110
  3.1.4.55.2  Elements .............................................................111
    3.1.4.55.2.1  SearchDatabaseCleanup .....................................111
    3.1.4.55.2.2  SearchDatabaseCleanupResponse ........................111
3.1.4.56  SetContentSourcesMetadata ..........................................111
  3.1.4.56.1  Messages ...............................................................112
    3.1.4.56.1.1  SetContentSourcesMetadataSoapIn .......................112
    3.1.4.56.1.2  SetContentSourcesMetadataSoapOut .....................112
  3.1.4.56.2  Elements .............................................................112
    3.1.4.56.2.1  SetContentSourcesMetadata ..............................112
    3.1.4.56.2.2  SetContentSourcesMetadataResponse ....................113
3.1.4.57  SetCrawlRuleCredentials ...........................................113
  3.1.4.57.1  Messages ...............................................................114
    3.1.4.57.1.1  SetCrawlRuleCredentialsSoapIn ............................114
    3.1.4.57.1.2  SetCrawlRuleCredentialsSoapOut .........................114
  3.1.4.57.2  Elements .............................................................114
    3.1.4.57.2.1  SetCrawlRuleCredentials ....................................114
    3.1.4.57.2.2  SetCrawlRuleCredentialsResponse ........................116
3.1.4.58  SetCrawlRulePriority ................................................117
  3.1.4.58.1  Messages ...............................................................117
    3.1.4.58.1.1  SetCrawlRulePrioritySoapIn ................................117
    3.1.4.58.1.2  SetCrawlRulePrioritySoapOut ................................118
  3.1.4.58.2  Elements .............................................................118
    3.1.4.58.2.1  SetCrawlRulePriority ........................................118
    3.1.4.58.2.2  SetCrawlRulePriorityResponse ................................118
3.1.4.59  SetDefaultGatheringAccount ......................................119
  3.1.4.59.1  Messages ...............................................................119
    3.1.4.59.1.1  SetDefaultGatheringAccountSoapIn ........................119
    3.1.4.59.1.2  SetDefaultGatheringAccountSoapOut .......................119
  3.1.4.59.2  Elements .............................................................120
    3.1.4.59.2.1  SetDefaultGatheringAccount ............................120
    3.1.4.59.2.2  SetDefaultGatheringAccountResponse ....................120
3.1.4.60  SetDefaultImportAccount ...........................................120
  3.1.4.60.1  Messages ...............................................................121
    3.1.4.60.1.1  SetDefaultImportAccountSoapIn ............................121
    3.1.4.60.1.2  SetDefaultImportAccountSoapOut ........................121
  3.1.4.60.2  Elements .............................................................121
    3.1.4.60.2.1  SetDefaultImportAccount ....................................121
    3.1.4.60.2.2  SetDefaultImportAccountResponse ........................122
3.1.4.61  SetImportAccount .....................................................122
  3.1.4.61.1  Messages ...............................................................123
    3.1.4.61.1.1  SetImportAccountSoapIn ....................................123
    3.1.4.61.1.2  SetImportAccountSoapOut ..................................123
  3.1.4.61.2  Elements .............................................................123
    3.1.4.61.2.1  SetImportAccount ............................................123
    3.1.4.61.2.2  SetImportAccountResponse ................................124
3.1.4.62  SetIsExtensionIncludeList ........................................124
  3.1.4.62.1  Messages ...............................................................124
3.1.4.62.1.1 SetLastErrorExtensionIncludeListSoapIn ........................................124
3.1.4.62.1.2 SetLastErrorExtensionIncludeListSoapOut ..................................125
3.1.4.62.2 Elements .................................................................................125
3.1.4.62.2.1 SetLastErrorExtensionIncludeList .............................................125
3.1.4.62.2.2 SetLastErrorExtensionIncludeListResponse ................................125
3.1.4.63 SetRetryLimit .............................................................................126
3.1.4.63.1 Messages .................................................................................126
3.1.4.63.1.1 SetRetryLimitSoapIn .................................................................126
3.1.4.63.1.2 SetRetryLimitSoapOut ...............................................................126
3.1.4.63.2 Elements .................................................................................126
3.1.4.63.2.1 SetRetryLimit ........................................................................127
3.1.4.63.2.2 SetRetryLimitResponse ............................................................127
3.1.4.64 StartCrawl ....................................................................................127
3.1.4.64.1 Messages ...............................................................................128
3.1.4.64.1.1 StartCrawlSoapIn .................................................................128
3.1.4.64.1.2 StartCrawlSoapOut .................................................................128
3.1.4.64.2 Elements ...............................................................................128
3.1.4.64.2.1 StartCrawl ........................................................................128
3.1.4.64.2.2 StartCrawlResponse ...............................................................129
3.1.4.65 StartFullImport+ ..........................................................................129
3.1.4.65.1 Messages ...............................................................................129
3.1.4.65.1.1 StartFullImportSoapIn .............................................................129
3.1.4.65.1.2 StartFullImportSoapOut ..........................................................130
3.1.4.65.2 Elements ...............................................................................130
3.1.4.65.2.1 StartFullImport ................................................................130
3.1.4.65.2.2 StartFullImportResponse .......................................................130
3.1.4.66 StartIncrementalImport .................................................................131
3.1.4.66.1 Messages ...............................................................................131
3.1.4.66.1.1 StartIncrementalImportSoapIn .................................................131
3.1.4.66.1.2 StartIncrementalImportSoapOut .............................................131
3.1.4.66.2 Elements ...............................................................................131
3.1.4.66.2.1 StartIncrementalImport ........................................................131
3.1.4.66.2.2 StartIncrementalImportResponse ........................................132
3.1.4.67 StartRankingUpdate ......................................................................132
3.1.4.67.1 Messages ...............................................................................132
3.1.4.67.1.1 StartRankingUpdateSoapIn ......................................................132
3.1.4.67.1.2 StartRankingUpdateSoapOut ..................................................133
3.1.4.67.2 Elements ...............................................................................133
3.1.4.67.2.1 StartRankingUpdate ...............................................................133
3.1.4.67.2.2 StartRankingUpdateResponse ...............................................133
3.1.4.67.3 Complex Types ......................................................................133
3.1.4.67.4 Simple Types .........................................................................133
3.1.4.67.4.1 RankingUpdateType ..............................................................133
3.1.4.68 StopCrawl ...................................................................................134
3.1.4.68.1 Messages ...............................................................................134
3.1.4.68.1.1 StopCrawlSoapIn .................................................................134
3.1.4.68.1.2 StopCrawlSoapOut ...............................................................135
3.1.4.68.2 Elements ...............................................................................135
3.1.4.68.2.1 StopCrawl ........................................................................135
3.1.4.68.2.2 StopCrawlResponse .............................................................135
3.1.4.69 StopImport ................................................................................135
3.1.4.69.1 Messages ...............................................................................136
3.1.4.69.1.1 StopImportSoapIn ...............................................................136
3.1.4.69.1.2 StopImportSoapOut .............................................................136
3.1.4.69.2 Elements ...............................................................................136
3.1.4.69.2.1 StopImport ........................................................................136
3.1.4.69.2.2 StopImportResponse ............................................................137
3.1.4.70 SystemDrive .............................................................................137


3.1.4.71 TestCrawlRule

3.1.4.71.1 Messages

3.1.4.71.1.1 TestCrawlRuleSoapIn

3.1.4.71.1.2 TestCrawlRuleSoapOut

3.1.4.71.2 Elements

3.1.4.71.2.1 TestCrawlRule

3.1.4.71.2.2 TestCrawlRuleResponse

3.1.4.72 TestCrawlRules

3.1.4.72.1 Messages

3.1.4.72.1.1 TestCrawlRulesSoapIn

3.1.4.72.1.2 TestCrawlRulesSoapOut

3.1.4.72.2 Elements

3.1.4.72.2.1 TestCrawlRules

3.1.4.72.2.2 TestCrawlRulesResponse

3.1.4.73 UpdateCrawlRule

3.1.4.73.1 Messages

3.1.4.73.1.1 UpdateCrawlRuleSoapIn

3.1.4.73.1.2 UpdateCrawlRuleSoapOut

3.1.4.73.2 Elements

3.1.4.73.2.1 UpdateCrawlRule

3.1.4.73.2.2 UpdateCrawlRuleResponse

3.1.4.74 ValidateApplicationPath

3.1.4.74.1 Messages

3.1.4.74.1.1 ValidateApplicationPathSoapIn

3.1.4.74.1.2 ValidateApplicationPathSoapOut

3.1.4.74.2 Elements

3.1.4.74.2.1 ValidateApplicationPath

3.1.4.74.2.2 ValidateApplicationPathResponse

3.1.4.75 ValidatePath

3.1.4.76 ValidateScheduleTrigger

3.1.4.76.1 Messages

3.1.4.76.1.1 ValidateScheduleTriggerSoapIn

3.1.4.76.1.2 ValidateScheduleTriggerSoapOut

3.1.4.76.2 Elements

3.1.4.76.2.1 ValidateScheduleTrigger

3.1.4.76.2.2 ValidateScheduleTriggerResponse

3.1.4.76.3 Complex Types

3.1.4.76.3.1 TriggerUpdate

3.1.5 Timer Events

3.1.6 Other Local Events

4 Protocol Examples

4.1 Create and Configure a Content Source

4.1.1 GetVersion

4.1.2 GetContentSources

4.1.3 ValidateScheduleTrigger

4.1.4 AddContentSource

4.1.5 EditContentSource

4.2 Create a Crawl Rule

4.2.1 GetCrawlRuleList

4.2.2 AddCrawlRule

4.2.3 UpdateCrawlRule

4.2.4 SetCrawlRuleCredentials

4.3 Crawl a Content Source

4.3.1 GetContentSources

4.3.2 StartCrawl

4.4 User Profile Import

4.4.1 SetImportAccount

12 / 223

[MS-ADMWS] - v20171212
Search Service Administration Web Service Protocol
Copyright © 2017 Microsoft Corporation
Release: December 12, 2017
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2</td>
<td>StartFullImport</td>
<td>155</td>
</tr>
<tr>
<td>4.5</td>
<td>Failure Example</td>
<td>156</td>
</tr>
<tr>
<td>4.5.1</td>
<td>GetContentSources</td>
<td>156</td>
</tr>
<tr>
<td>5</td>
<td>Security</td>
<td>160</td>
</tr>
<tr>
<td>5.1</td>
<td>Security Considerations for Implementers</td>
<td>160</td>
</tr>
<tr>
<td>5.2</td>
<td>Index of Security Parameters</td>
<td>160</td>
</tr>
<tr>
<td>6</td>
<td>Appendix A: Full WSDL</td>
<td>161</td>
</tr>
<tr>
<td>7</td>
<td>Appendix B: Product Behavior</td>
<td>218</td>
</tr>
<tr>
<td>8</td>
<td>Change Tracking</td>
<td>219</td>
</tr>
<tr>
<td>9</td>
<td>Index</td>
<td>220</td>
</tr>
</tbody>
</table>
1 Introduction

This document specifies the Search Service Administration Web Service Protocol. This protocol enables the protocol client to configure and control the indexing functions of the search system. It allows the protocol client to define the scope of indexing by configuring starting points and restriction rules, credentials used during indexing by the protocol server, and control when the indexing process takes place.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

access URL: An internal Uniform Resource Locator (URL) that is used by a crawler to identify and gain access to an item.

Active Directory: A general-purpose network directory service. Active Directory also refers to the Windows implementation of a directory service. Active Directory stores information about a variety of objects in the network. User accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in Active Directory. Active Directory is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Kerberos, and DNS.

anchor content source: A content source that is used to import the anchor text from links between items into the full-text index catalog.

anchor crawl: A process in which anchor text from links between items is added to a full-text index catalog.

anonymous authentication: An authentication mode in which neither party verifies the identity of the other party.

authority hops: The number of site levels to be navigated from a start address to a specific item.

binary large object (BLOB): A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.

Business Data Connectivity (BDC): A shared service that stores information about business application data that exists outside a server farm. It can be used to display business data in lists, Web Parts, search results, user profiles, and custom applications. Previously referred to as Business Data Catalog.

certificate: A certificate is a collection of attributes and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for authentication and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [RFC3280] and [X509] sections 7 and 8.

configuration database: A database that is stored on a back-end database server and contains both persisted objects and site collection metadata for lookup purposes.
**content source**: A set of options for specifying the type of content to be crawled and the start addresses for the content to be indexed. A content source is defined by the protocol handler that is used to access specific systems, such as SharePoint sites, file systems, and external websites. A content source can contain up to 500 start addresses.

**cookie**: A small data file that is stored on a user's computer and carries state information between participating protocol servers and protocol clients.

**crawl**: The process of traversing a URL space to acquire items to record in a search catalog.

**crawl account**: A user account that has access to all of the content that is traversed by a crawl component.

**crawl mapping**: A mapping that associates an access URL, which is used to obtain an item from a content source, and a display URL, which is the address of the item.

**crawl queue**: A data structure that stores the list of items to crawl next.

**crawl rule**: A set of preferences that applies to a specific URL or range of URLs. A crawl rule can be used to include or exclude items in a crawl and to specify the content access account to use when crawling that URL or range of URLs.

**crawler**: A process that browses and indexes content from a content source.

**delete crawl**: A process that is started automatically after a content source or start address deletion occurs and removes associated items from a search catalog.

**directory service (DS)**: A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also Active Directory.

**display URL**: The URL that is displayed on a search results page for each search result. This can be different than an access URL. See also access URL.

**domain**: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a domain controller (DC) and host a member list that identifies all members of the domain, as well as optionally hosting the Active Directory service. The domain controller provides authentication of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

**domain name**: The name given by an administrator to a collection of networked computers that share a common directory. Part of the domain naming service naming structure, domain names consist of a sequence of name labels separated by periods.

**drive letter**: One of the 26 alphabetical characters A-Z, in uppercase or lowercase, that is assigned to a volume. Drive letters serve as a namespace through which data on the volume can be accessed. A volume with a drive letter can be referred to with the drive letter followed by a colon (for example, C:).

**exclusion list**: A list of items to exclude from query results and to remove from a search index the next time that a crawl occurs.

**file**: A single, discrete unit of content.

**folder**: A file system construct. File systems organize a volume's data by providing a hierarchy of objects, which are referred to as folders or directories, that contain files and can also contain other folders.

**forms authentication**: An authentication method in which protocol clients redirect unauthenticated requests to an HTML form by using HTTP. If the protocol client authenticates
the request, the system issues a **cookie** that stores the credentials or a key for reacquiring the identity. In subsequent requests, the cookie is submitted in request headers and the requests are authenticated and authorized by an ASP.NET event handler that uses the validation method that is specified by the protocol client.

**full crawl**: A crawl process that indexes all of the items in a specified content source, regardless of whether the item was modified.

**full-text index catalog**: A collection of full-text index components and other files that are organized in a specific directory structure and contain the data that is needed to perform queries.

**globally unique identifier (GUID)**: A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the GUID. See also universally unique identifier (UUID).

**host hop**: The process of traversing to a server with a different host name during a crawl.

**HRESULT**: An integer value that indicates the result or status of an operation. A particular HRESULT can have different meanings depending on the protocol using it. See [MS-ERREF] section 2.1 and specific protocol documents for further details.

**HTTP GET**: An HTTP method for retrieving a resource, as described in [RFC2616].

**HTTP POST**: An HTTP method, as described in [RFC2616].

**inclusion list**: A list of items to include in query results and to add to a search index the next time that a crawl occurs.

**incremental crawl**: A crawl process that includes logic to index only a subset of the items in a content source that is crawled based on item modifications.

**index server**: A server that is assigned the task of crawling.

**item**: A unit of content that can be indexed and searched by a search application.

**metadata index**: A data structure that is stored on a back-end database server. It stores properties that are associated with each item, and the attributes of those properties.

**page hop**: The process of traversing from one item to another during a crawl. See also site hop.

**portal content project**: A primary search catalog that contains all of the content sources and settings for an administrator-defined crawl.

**query component**: A portion of a URL that follows a question mark (?), as described in [RFC3986].

**query server**: A server that has been assigned the task of fulfilling search queries.

**search catalog**: All of the crawl data that is associated with a specific search application. A search catalog provides information that is used to generate query results.

**search database**: A database that stores search-related information, including stored procedures and tables that are used for crawl data, document metadata, and administration information.

**search query**: A complete set of conditions that are used to generate search results, including query text, sort order, and ranking parameters.
**search service account**: A user account under which a search service runs.

**security trimmer**: A filter that is used to limit search results to only those resources that a user can view, based on the user's permission level and the access control list (ACL) for a resource. A security trimmer helps to ensure that search results display only those resources that a user has permission to view.

**share**: A resource offered by a Common Internet File System (CIFS) server for access by CIFS clients over the network. A share typically represents a directory tree and its included files (referred to commonly as a "disk share" or "file share") or a printer (a "print share"). If the information about the share is saved in persistent store (for example, Windows registry) and reloaded when a file server is restarted, then the share is referred to as a "sticky share". Some share names are reserved for specific functions and are referred to as special shares: IPC$, reserved for interprocess communication, ADMIN$, reserved for remote administration, and A$, B$, C$ (and other local disk names followed by a dollar sign), assigned to local disk devices.

**Shared Services Provider (SSP)**: A logical grouping of shared service applications, and their supporting resources, that can be configured and managed from a single server and can be used by multiple server farms.

**site**: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

**SOAP action**: The HTTP request header field used to indicate the intent of the SOAP request, using a URI value. See [SOAP1.1] section 6.1.1 for more information.

**SOAP body**: A container for the payload data being delivered by a SOAP message to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

**SOAP fault**: A container for error and status information within a SOAP message. See [SOAP1.2-1/2007] section 5.4 for more information.

**start address**: A URL that identifies a point at which to start a crawl. Administrators specify start addresses when they create or edit a content source.

**Uniform Resource Identifier (URI)**: A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

**Uniform Resource Locator (URL)**: A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

**Universal Naming Convention (UNC)**: A string format that specifies the location of a resource. For more information, see [MS-DTYP] section 2.2.57.

**URL encode**: The process of encoding characters that have reserved meanings for a Uniform Resource Locator (URL), as described in [RFC1738].

**URL space**: A list of Uniform Resource Locators (URLs) that contains information about the links from each URL to other URLs.

**user profile import**: The process of importing records from a directory service (DS) to a user profile store.

**web application**: A container in a configuration database that stores administrative settings and entry-point URLs for site collections.

**web server**: A server computer that hosts websites and responds to requests from applications.

**web service**: A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as HTTP, Simple Mail Transfer
Protocol (SMTP), or File Transfer Protocol (FTP). Web services can perform functions that range from simple requests to complicated business processes.

**Web Services Description Language (WSDL):** An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

**website:** (1) A group of related webpages that is hosted by a server on the World Wide Web or an intranet. Each website has its own entry points, metadata, administration settings, and workflows. Also referred to as site.

(2) A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and site.

**WSDL message:** An abstract, typed definition of the data that is communicated during a WSDL operation [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

**X.509:** An ITU-T standard for public key infrastructure subsequently adapted by the IETF, as specified in [RFC3280].

**XML namespace:** A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

**XML namespace prefix:** An abbreviated form of an XML namespace, as described in [XML].

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

### 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

#### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MS-NRTP] Microsoft Corporation, ".NET Remoting: Core Protocol".


1.3 Protocol Overview (Synopsis)

The Search Service Administration Web Service protocol enables a protocol client to configure and control the crawl function of the search service’s index server.

A crawl process involves traversal of the URL space by discovering and following the links from items. Before a crawl can be started, the URL space is defined by specifying a set of start address URLs and a set of rules that prevent the crawl from following links beyond desired boundaries of the potentially infinite unrestricted URL space, for example to prevent it from crawling the Internet.

During a crawl, the index server typically produces a set of data structures, such as the full-text index catalog and the metadata index that can be used to efficiently serve search queries.

Another specialized application of the crawl process is to perform a user profile import.

A typical scenario for using this protocol involves the following steps:

- Configuring one or more content sources containing at least one start address.
- Configuring zero or more crawl rules.
- Starting a crawl.
- Configuring a user profile import by specifying one or more user profile import accounts.
- Starting a user profile import.
1.4 Relationship to Other Protocols

This protocol uses the SOAP message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits those messages by using HTTP, as described in [RFC2616], or Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS), as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol:

![Diagram showing the relationship between Search Service Administration Web Service Protocol and other protocols]

**Figure 1: This protocol in relation to other protocols**

1.5 Prerequisites/Preconditions

This protocol operates against a Web site (2) that is identified by a URL that is known by protocol clients. The protocol server endpoint is formed by concatenating the URL of the Web site with an ending ‘/’ character, the name of the SSP, and ”/Search/SearchAdmin.asmx” together, for example http://www.contoso.com:56737/SharedServices1/Search/SearchAdmin.asmx.

This protocol assumes that authentication has been performed by the underlying protocols.

1.6 Applicability Statement

This protocol is designed for configuring index server with not more than 500 content sources and 100 crawl rules.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP as specified in section 2.1.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section 2.2 and section 3.1.4.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.
2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty, null, or not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty, not null, and present**.

2.1 Transport

Protocol servers MUST support SOAP over HTTP. Protocol servers MAY additionally support SOAP over HTTPS for securing communication with clients.

Protocol messages MUST be formatted as specified either in [SOAP1.1], section 4, or in [SOAP1.2/1], section 5. Protocol server faults MUST be returned either using HTTP Status Codes as specified in [RFC2616], section 10, or using SOAP faults as specified either in [SOAP1.1], section 4.4 or in [SOAP1.2/1], section 5.4. The version of the SOAP fault returned MUST correspond to the version of SOAP used for the request **WSDL message**.

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses XML Schema as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and WSDL as defined in [WSDL].

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [XMLNS]. Although this specification associates a specific **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1]</td>
</tr>
<tr>
<td>soap12</td>
<td><a href="http://schemas.xmlsoap.org/wsd/soap12/">http://schemas.xmlsoap.org/wsd/soap12/</a></td>
<td>[SOAP1.2/1] [SOAP1.2/2]</td>
</tr>
<tr>
<td>mime</td>
<td><a href="http://schemas.xmlsoap.org/wsd/mime/">http://schemas.xmlsoap.org/wsd/mime/</a></td>
<td></td>
</tr>
<tr>
<td>soap</td>
<td><a href="http://schemas.xmlsoap.org/wsd/soap/">http://schemas.xmlsoap.org/wsd/soap/</a></td>
<td>[SOAP1.1]</td>
</tr>
<tr>
<td>tm</td>
<td><a href="http://microsoft.com/wsd/mime/textMatching/">http://microsoft.com/wsd/mime/textMatching/</a></td>
<td></td>
</tr>
<tr>
<td>soapenc</td>
<td><a href="http://schemas.xmlsoap.org/soap/encoding/">http://schemas.xmlsoap.org/soap/encoding/</a></td>
<td></td>
</tr>
<tr>
<td>wsd1</td>
<td><a href="http://schemas.xmlsoap.org/wsd/">http://schemas.xmlsoap.org/wsd/</a></td>
<td>[WSDL]</td>
</tr>
</tbody>
</table>
2.2.2 Messages

The operations defined in section 3.1.4 MAY return a SOAP fault. A SOAP fault element is used to carry error and status information within a SOAP message when there is API misuse or where there are configuration failures.

2.2.3 Elements

This specification does not define any common XML Schema element definitions.

2.2.4 Complex Types

The following table summarizes the set of common XML Schema complex type definitions defined by this specification. XML Schema complex type definitions that are specific to a particular operation are described with the operation.

<table>
<thead>
<tr>
<th>Complex Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayOfString</td>
<td>An array of arbitrary string values.</td>
</tr>
<tr>
<td>ContentSourceDynamicPropsInternal</td>
<td>The crawl status of a content source with respect to the most recent crawl.</td>
</tr>
<tr>
<td>CrawlRuleInternal</td>
<td>The crawl rule properties.</td>
</tr>
<tr>
<td>ContentSourceInternal</td>
<td>The content source properties configured by the protocol client.</td>
</tr>
</tbody>
</table>

2.2.4.1 ArrayOfString

The **ArrayOfString** type represents an array of arbitrary string values.

```xml
<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element name="string" type="s:string" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

**string**: A single string value.

2.2.4.2 ContentSourceDynamicPropsInternal

The **ContentSourceDynamicPropsInternal** type represents crawl status information for a content source.

```xml
<s:complexType name="ContentSourceDynamicPropsInternal">
  <s:sequence>
    <s:element name="crawlStatus" type="s:int"/>
    <s:element name="crawlStarted" type="s:dateTime"/>
    <s:element name="crawlCompleted" type="s:dateTime"/>
    <s:element name="errorCount" type="s:int"/>
    <s:element name="crawlSuccesses" type="s:int"/>
  </s:sequence>
</s:complexType>
```

**crawlStatus**: The crawl status. MUST be one of the following values:
<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No crawl is in progress.</td>
</tr>
<tr>
<td>1</td>
<td>A full crawl is in progress.</td>
</tr>
<tr>
<td>2</td>
<td>The crawl is paused, and no further activity will take place for the current crawl until the crawl is resumed.</td>
</tr>
<tr>
<td>4</td>
<td>The crawl is resuming after system startup.</td>
</tr>
<tr>
<td>5</td>
<td>The crawler application is shutting down, because the search service was stopped.</td>
</tr>
<tr>
<td>6</td>
<td>An incremental crawl is in progress.</td>
</tr>
<tr>
<td>8</td>
<td>The crawl is being stopped. The request to stop the crawl has been received, but the crawl hasn’t stopped yet.</td>
</tr>
<tr>
<td>9</td>
<td>The crawl is being paused. The request to pause the crawl has been received, but the crawl hasn’t paused yet.</td>
</tr>
</tbody>
</table>

**crawlStarted**: The start time for the most recent crawl. If no crawls have been started for the content source yet, this element MUST contain the value of 1899-12-30T00:00:00.

**crawlCompleted**: The end time of the last finished crawl. If the content source has not been crawled, this element MUST contain the value of 1899-12-30T00:00:00.

**errorCount**: The number of items that the crawler failed to get crawl during the most recent crawl. **errorCount** MUST be greater than or equal to 0. If no crawls have been performed on the content source, the number MUST be 0.

**crawlSuccesses**: The number of items successfully crawled during the most recent crawl of the content source. **crawlSuccesses** MUST be greater than or equal to 0. If no crawls have been performed on the content source, the number MUST be 0.

### 2.2.4.3 CrawlRuleInternal

The **CrawlRuleInternal** type represents the properties for a crawl rule.

```xml
<s:complexType name="CrawlRuleInternal">
  <s:sequence>
    <s:element name="path" type="s:string" minOccurs="0"/>
    <s:element name="type" type="s:int"/>
    <s:element name="authenticationType" type="s:int"/>
    <s:element name="accountName" type="s:string" minOccurs="0"/>
    <s:element name="contentClass" type="s:string" minOccurs="0"/>
    <s:element name="suppressIndexing" type="s:boolean"/>
    <s:element name="followComplexUrls" type="s:boolean"/>
    <s:element name="crawlAsHttp" type="s:boolean"/>
    <s:element name="enabled" type="s:boolean"/>
    <s:element name="pluggableSecurityTrimmerId" type="s:int"/>
    <s:element name="authUrl" type="s:string" minOccurs="0"/>
    <s:element name="authData" type="s:string" minOccurs="0"/>
    <s:element name="miscData" type="tns:ArrayOfString" minOccurs="0"/>
    <s:element name="accountLastModified" type="s:dateTime"/>
  </s:sequence>
</s:complexType>
```
**path:** A crawl rule path expression. MUST be present, and the length MUST be greater than 0 and less than 2048 characters. MUST be either a **Universal Naming Convention (UNC)** path or a URL, with the following characters allowed: `*` and `?`.

**type:** The crawl rule type. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Inclusion rule. URLs matching the <strong>path</strong> are included in the crawl.</td>
</tr>
<tr>
<td>1</td>
<td>Exclusion rule. URLs matching the <strong>path</strong> are not included in the crawl.</td>
</tr>
</tbody>
</table>

**authenticationType:** The authentication type for accessing matching URLs. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default access.</td>
</tr>
<tr>
<td>1</td>
<td>Integrated authentication.</td>
</tr>
<tr>
<td>2</td>
<td>Basic authentication. The user name and password are required for this authentication type.</td>
</tr>
<tr>
<td>3</td>
<td>Authentication using a <strong>certificate</strong>. A valid client certificate name is required for this authentication type.</td>
</tr>
<tr>
<td>4</td>
<td><strong>forms authentication</strong>. A valid URL for <strong>HTTP POST</strong> or <strong>HTTP GET</strong>, public and private parameters, and a list of error pages are used by this authentication type.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Cookie</strong> based authentication. Private parameters and a list of error pages are used by this authentication type.</td>
</tr>
</tbody>
</table>

Default access implies integrated authentication using credentials of the default **crawl account** for the crawler application.

**accountName:** If present, the length MUST be less than 256 characters. This element MUST be interpreted differently depending on the value of **authenticationType**. The following table specifies interpretation and restrictions for this element:

<table>
<thead>
<tr>
<th>authenticationType value</th>
<th>accountName interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>MUST be ignored.</td>
</tr>
<tr>
<td>1</td>
<td>The user name for integrated authentication.</td>
</tr>
<tr>
<td>2</td>
<td>The user name for basic authentication.</td>
</tr>
<tr>
<td>3</td>
<td>The <strong>X.509</strong> certificate name.</td>
</tr>
<tr>
<td>4</td>
<td>MUST be ignored.</td>
</tr>
<tr>
<td>5</td>
<td>MUST be ignored.</td>
</tr>
</tbody>
</table>

**contentClass:** Arbitrary metadata for the crawl rule. If present, the length MUST be less than 1024 characters.

**suppressIndexing:** MUST be one of the following values:
<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>The protocol server MUST crawl any URLs that match the URL specified for the crawl rule’s path.</td>
</tr>
<tr>
<td>true</td>
<td>The protocol server MUST NOT crawl any URLs that match the URL specified for the crawl rule’s path.</td>
</tr>
</tbody>
</table>

**followComplexUrls**: Specifies the crawl behavior on the matched URLs with a query component. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>Matching URLs with a query component MUST be followed during the crawl.</td>
</tr>
<tr>
<td>true</td>
<td>Matching URLs with a query component MUST be discarded during the crawl.</td>
</tr>
</tbody>
</table>

**crawlAsHttp**: Specifies whether to use the HTTP protocol to crawl matching links with the http:scheme. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>The protocol server crawls matching URLs with http:scheme using any protocol available for the repository.</td>
</tr>
<tr>
<td>true</td>
<td>The protocol server crawls matching URLs with http:scheme using the HTTP protocol.</td>
</tr>
</tbody>
</table>

**enabled**: Specifies whether the crawl rule is enabled or disabled. Disabled crawl rules are ignored by the protocol server. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>The crawl rule is disabled.</td>
</tr>
<tr>
<td>true</td>
<td>The crawl rule is enabled.</td>
</tr>
</tbody>
</table>

**pluggableSecurityTrimmerId**: The security trimmer identifier.

**authUrl**: If authenticationType is 4, authUrl contains the URL for the forms authentication type. If present, the length MUST be less than 2048 characters. The forms authentication type implies an HTTP GET or an HTTP POST to the authentication URL, as specified in [RFC2616], to obtain the authorization cookie. If authenticationType is set to 0, 1, 2, 3 or 5, this element MUST not be interpreted.

**authData**: If authenticationType is 4, this element represents the public authentication parameters for authUrl, according to the format of the HTTP form in authUrl. If authenticationType is set to 0, 1, 2, 3 or 5, this element MUST not be interpreted.

**miscData**: If authenticationType is set to 4 or to 5, this element represents a collection of error page URLs. When requests to retrieve content get redirected by item repositories during a crawl, these error page URLs are used to determine authentication errors. If authenticationType is set to 0, 1, 2 or 3, this element MUST not be interpreted.

**accountLastModified**: The latest time credentials were set or modified for this rule.
2.2.4.4 ContentSourceInternal

The **ContentSourceInternal** type represents the properties and status information for a content source.

```xml
<s:complexType name="ContentSourceInternal">
  <s:sequence>
    <s:element name="id" type="s:int"/>
    <s:element name="systemCreated" type="s:boolean"/>
    <s:element name="type" type="s:int"/>
    <s:element name="name" type="s:string" minOccurs="0"/>
    <s:element name="wssCrawlStyle" type="s:int"/>
    <s:element name="metadata" type="s:string" minOccurs="0"/>
    <s:element name="followDirectories" type="s:boolean"/>
    <s:element name="pageDepth" type="s:int"/>
    <s:element name="siteDepth" type="s:int"/>
    <s:element name="startAddresses" type="tns:ArrayOfString" minOccurs="0"/>
    <s:element name="throttleStart" type="s:int"/>
    <s:element name="throttleDuration" type="s:int"/>
    <s:element name="fullCrawlTrigger" type="s:base64Binary" minOccurs="0"/>
    <s:element name="incCrawlTrigger" type="s:base64Binary" minOccurs="0"/>
    <s:element name="nextIncCrawlStart" type="s:dateTime"/>
    <s:element name="nextFullCrawlStart" type="s:dateTime"/>
    <s:element name="incScheduleDescription" type="s:string" minOccurs="0"/>
    <s:element name="fullScheduleDescription" type="s:string" minOccurs="0"/>
    <s:element name="dynamic" type="tns:ContentSourceDynamicPropsInternal" minOccurs="0"/>
  </s:sequence>
</s:complexType>
```

**id**: The content source identifier. The identifier MUST be greater than 0.

**systemCreated**: Specifies whether the content source was created automatically by the system or if the protocol client created it using this protocol. MUST be one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>The content source was automatically created after server installation.</td>
</tr>
<tr>
<td>False</td>
<td>The content source was created by the protocol client using this protocol.</td>
</tr>
</tbody>
</table>

**type**: The content source type. MUST be one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Web sites</td>
</tr>
<tr>
<td>1</td>
<td>Sites</td>
</tr>
<tr>
<td>2</td>
<td>Lotus Notes database</td>
</tr>
<tr>
<td>3</td>
<td>File shares</td>
</tr>
<tr>
<td>4</td>
<td>Exchange public folders</td>
</tr>
<tr>
<td>5</td>
<td>Custom</td>
</tr>
<tr>
<td>6</td>
<td>Business Data Connectivity (BDC)</td>
</tr>
</tbody>
</table>

**name**: The content source name. MUST be present, and the length of the name MUST be greater than 0 characters and less than or equal to 255 characters.

**wssCrawlStyle**: The scope of start addresses. MUST be one of the following values.
### Value | Meaning
---|---
0 | All **Web applications** pointed to by the start addresses are to be crawled.
1 | Only sites pointed to by the start addresses are to be crawled.

**metadata**: An arbitrary custom string that the protocol client associated with the content source. If present, the length MUST be less than 1024 characters. The utilization of the metadata string depends on the protocol client implementation details.

**followDirectories**: Specifies how items are discovered by the crawler during the crawl. MUST be one of the following values.

| Value | Meaning |
---|---|
true | Items are discovered through **folder** links. |
false | Items are discovered through hyperlinks. |

**pageDepth**: The maximum number of **page hops** allowed when this content source is crawled. The number MUST be greater than or equal to 0. If there is no limit, the number MUST be 2147483647.

**siteDepth**: The maximum number of **host hops** allowed when this content source is crawled. The number MUST be greater than or equal to 0. If there is no limit, the number MUST be 2147483647.

**startAddresses**: An array of zero or more start addresses. This element MUST be present. Each element in the array MUST contain a single **URL** or **UNC** path of a start address, and the length MUST be less than 2048 characters.

**throttleStart**: This element is unused. **throttleStart** SHOULD contain 0 and MUST be ignored by the protocol server.

**throttleDuration**: This element is unused. **throttleDuration** SHOULD contain 0 and MUST be ignored by the protocol server.

**fullCrawlTrigger**: If this element is present, it MUST contain a trigger as defined in [MS-TSCH], section 2.4.2.11 for the **full crawl** schedule. If this element is not present, it means that the trigger is not defined.

**incCrawlTrigger**: If this element is present, it MUST contain a trigger as defined in [MS-TSCH], 2.4.2.11 for the **incremental crawl** schedule. If this element is not present, it means that the trigger is not defined.

**nextIncCrawlStart**: The start time of the next scheduled incremental crawl. If there is no incremental crawl scheduled, this element MUST contain the value 9999-12-31T23:59:59.9999999.

**nextFullCrawlStart**: The start time of the next scheduled full crawl. If there is no full crawl scheduled, this element MUST contain the value 9999-12-31T23:59:59.9999999.

**incScheduleDescription**: The text description for an incremental crawl schedule. If the **incCrawlTrigger** element is present, this element MUST also be present. If the **incCrawlTrigger** element is not present, this element MUST NOT be present.

**fullScheduleDescription**: The text description for a full crawl schedule. If the **fullCrawlTrigger** element is present, this element MUST also be present. If the **fullCrawlTrigger** element is not present, this element MUST NOT be present.

**dynamic**: This element MUST be present and MUST contain the content source status information as specified in section 2.2.4.2.
2.2.5 Simple Types
This specification does not define any common XML Schema simple type definitions.

2.2.6 Attributes
This specification does not define any common XML Schema attribute definitions.

2.2.7 Groups
This specification does not define any common XML Schema group definitions.

2.2.8 Attribute Groups
This specification does not define any common XML Schema attribute group definitions.

2.2.9 Exception Types
This section specifies exceptions that are used by this protocol. The exception definitions in this section use the notation specified in [MS-NRTP], section 2.2.5. The instructions to map the exceptions to the SOAP Format [SOAP1.1] are specified in [SOAP1.1], section 4.4 or in [SOAP1.2], section 5.4.

2.2.9.1 ConcurrencyException
The ConcurrencyException exception type is a derived class of System.Exception as specified in [MS-NRTP] section 2.2.7.

The type name is "Microsoft.Office.Server.Search.Administration". The library name is "Microsoft.Office.Server.Search, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c". There are no members other than those inherited from the System.Exception class.

```csharp
{
    public class ConcurrencyException : System.Exception
    {
    }
}
```

A ConcurrencyException is thrown when multiple protocol clients send conflicting requests to the protocol server.

2.2.9.2 UpdatedConcurrencyException
The UpdatedConcurrencyException exception type is a derived class of the ConcurrencyException class as specified in section 2.2.9.1.

The type name is "Microsoft.Office.Server.Search.Administration". The library name is "Microsoft.Office.Server.Search, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c". There are no members other than those inherited from the ConcurrencyException class.
An **UpdatedConcurrencyException** is thrown when the protocol server receives conflicting update requests from multiple protocol clients.

### 2.2.9.3 DeletedConcurrencyException

The **DeletedConcurrencyException** exception type is a derived class of **ConcurrentException** as specified in section **2.2.9.1**.

The type name is "Microsoft.Office.Server.Search.Administration". The library name is "Microsoft.Office.Server.Search, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c".

There are no members other than those inherited from the **ConcurreuntyException** class.

```csharp
{
    {
    }
}
```

A **DeletedConcurrencyException** is thrown when protocol server receives a request to operate on an object already deleted.

### 2.2.9.4 ExternalException

The **ExternalException** exception type is a derived class of **SystemException** as specified in [MS-NRTP], section 2.2.2.8. The library name of the class is "mscorlib". There are no members other than those inherited from the **SystemException** class.

```csharp
Namespace System.Runtime.InteropServices
{
    Class ExternalException:System.SystemException
    {
    }
}
```

### 2.2.9.5 COMException

The **COMException** exception type is a derived class of **ExternalException** as specified in section **2.2.9.4** of this document. The library name of the class is "mscorlib". There are no members other than those inherited from the **ExternalException** class.

```csharp
Namespace System.Runtime.InteropServices
{
    Class COMException:System.Runtime.InteropServices.ComException
    {
    }
}
```
2.2.9.6 KeyNotFoundException

The KeyNotFoundException exception type is a derived class of SystemException as specified in [MS-NRTP], section 2.2.2.8. The library name of the class is "mscorlib". There are no members other than those inherited from the SystemException class. This class has an additional constraint that the HRESULT MUST be the hex value 0x80131577.

```
namespace System.Collections.Generic
{
    class KeyNotFoundException : System.SystemException
    {
    }
}
```

ParamName: The parameter name of the argument which caused this exception to be raised.

2.2.9.7 ArgumentException

The ArgumentException exception type is a derived class of SystemException as specified in [MS-NRTP], section 2.2.2.8. The library name of the class is "mscorlib". There is one member other than those inherited from the SystemException class. This class has an additional constraint that the HRESULT MUST be the hex value 0x80070057.

```
namespace System
{
    class ArgumentException : System.SystemException
    {
        string ParamName;
    }
}
```

2.2.9.8 InvalidEnumArgumentException

The InvalidEnumArgumentException exception type is a derived class of ArgumentException as specified in section 2.2.9.7 of this document. The library name of the class is "mscorlib". There are no members other than those inherited from the ArgumentException class.

```
namespace System
{
    class InvalidEnumArgumentException : System.ArgumentException
    {
    }
}
```

2.2.9.9 InvalidOperationException

The InvalidOperationException exception type is a derived class of SystemException as specified in [MS-NRTP] section 2.2.2.8. The library name of the class is "mscorlib". There are no members other than those inherited from the SystemException class. This class has an additional constraint that the HRESULT MUST be the hex value 0x80131509.

```
namespace System
{
    class InvalidOperationException : System.SystemException
    {
    }
}
```
namespace System
{
    class InvalidOperationException: System.SystemException
    {
    }
}
3 Protocol Details

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be empty, null, or not present but the behavior of the protocol as specified restricts the same elements to being non-empty, not null, and present.

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

Except where specified, protocol clients SHOULD interpret HTTP Status Codes returned by the protocol server as specified in [RFC2616], section 10.

This protocol allows protocol servers to notify protocol clients of application-level faults using SOAP faults. Except where specified, these SOAP faults are not significant for interoperability, and protocol clients can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults either using HTTP Status Codes or using SOAP faults as specified previously in this section.

This protocol allows protocol servers to perform implementation-specific localization of text in various messages. Except where specified, the localization of this text is an implementation-specific behavior of the protocol server and not significant for interoperability

3.1 SearchApplicationWebService Soap Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

3.1.1.1 Object hierarchy

The protocol server maintains a hierarchy of objects representing state of the Search Service Administration Web Service protocol. The properties of these objects affect the behavior of the index server during the crawl.
3.1.1.2 Crawler Application

This is the top level object of the hierarchy. There is one instance of the crawler application per SSP. Accordingly, there is one instance of the server endpoint for this protocol, the **Web service** URL.

**version:** The configuration version of this crawler application. The version is incremented whenever the application state is updated by the protocol client. Any method that updates the application state requires the last known configuration version from the protocol client. If the protocol client version does not match with the current application version, the operation fails. This will typically happen when multiple protocol clients try to update the application state at the same time.

**name:** A **GUID** that uniquely identifies the crawler application.
retryLimit: The number of times the index server will retry fetching the item from any content source, (for example, a Web server) during the crawl in case certain types of recoverable protocol errors occur, for example, timeouts.

defaultCrawlingAccount: The name of the default crawl account that the index server uses to authenticate itself during the crawl when no other account is specified in a matching crawl rule.

defaultCrawlingAccountPassword: The password required for authentication of the defaultCrawlingAccount.

lotusNotesConfigured: If true, the Lotus Notes configuration was complete. This is specific for crawl configuration of the Lotus Notes content source. The configuration details for a Lotus Notes content source is specific to the implementation of the index server, but as far as this protocol is concerned the protocol server MUST be aware when the Lotus Notes configuration took place and communicate this information to the protocol client.

listKnownLotusNotesServers: The list of known Lotus Notes server names. This is specific for crawl configuration for Lotus Notes content sources.

3.1.1.3 Portal Content Project

The crawler application contains exactly one instance of the portal content project. This object’s properties control the index server behavior and state during crawls for all types of content source except for the user profile content source.

extension: A list of file name extensions recognized by the index server. The index server parses all links discovered during the crawl and extracts the file extension from them. The file extension is checked against the extension list. Depending on the setting of isExtensionIncludeList property the link is discarded or added to the crawl queue.

isExtensionIncludeList: A Boolean value that specifies whether the list of file extensions in extension is an inclusion list or an exclusion list. If true, extension is an inclusion list; otherwise, extension is an exclusion list.

contentSourcesMetadata: A custom metadata string associated with the portal content project. The value of this property is ignored by the protocol server, but can be interpreted by the protocol client to associate arbitrary metadata with the collection of content sources.

clientCertificateNames: The list of names of X.509 client certificates used by crawl rules with authentication type of 3.

activeDocuments: The list of URLs of items currently being crawled by the index server.

pauseReason: A 32-bit number that contains the set of reasons why the search catalog is paused. Each bit indicates a particular reason. If there is more than one reason why the search catalog was paused, multiple bits corresponding to the reasons are turned on. The search catalog is resumed when all bits are turned off. MUST only turn on the following bits:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Paused because of certain unspecified internal event not initiated by the protocol client.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Paused because of certain unspecified internal event not initiated by the protocol client.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Paused for backup.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Paused for query server initialization.</td>
</tr>
</tbody>
</table>
### 3.1.1.4 Content Source

The portal content project contains a collection of zero or more content source objects. This object represents a content source that can be used to start a crawl on the index server.

**id:** The unique identifier of the content source in the collection. Assigned by the protocol server when a new content source is added.

**type:** The content source type. This type is used by the crawler as a hint to determine which technology to use to crawl the repository pointed to by the start addresses. **MUST** be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Web sites</td>
</tr>
<tr>
<td>1</td>
<td>Sites</td>
</tr>
<tr>
<td>2</td>
<td>Lotus Notes database</td>
</tr>
<tr>
<td>3</td>
<td>File shares</td>
</tr>
<tr>
<td>4</td>
<td>Exchange public folders</td>
</tr>
<tr>
<td>5</td>
<td>Custom</td>
</tr>
<tr>
<td>6</td>
<td><strong>Business Data Connectivity (BDC)</strong></td>
</tr>
</tbody>
</table>

**systemCreated:** If true, the content source was created during the initial system configuration and cannot be deleted by the protocol client. Any content sources added by the protocol client will have the **systemCreated** set to false.

**name:** The content source name. This is the label intended to be displayed in user interface.

**wssCrawlStyle:** The type of the crawl performed while crawling sites. If **0**, then the entire Web applications pointed to by start addresses are crawled. If **1**, then only the specific sites pointed by the start addresses are crawled without enumerating all sites in the Web application.

**metadata:** The arbitrary metadata associated by the protocol client with the content source. The value of this property is ignored by the protocol server, but can be interpreted by the protocol client to associate arbitrary metadata with the collection of content sources.

**followDirectories:** If true, only links provided by the repository being crawled are followed during the crawl, and links discovered within items are discarded. If false, only links discovered within items are followed.

**pageDepth:** The maximum permitted depth of the URL space traversal, including traversal within a single site or across different sites. Whenever a link is followed by the index server during the crawl the depth counter is incremented. The depth counter cannot increase beyond the **pageDepth** of the content source. For example, if the **pageDepth** is **1** and Page A links to Page B, which links to Pages C and D, then neither pages C nor D will be crawled because the depth counter would exceed **pageDepth**.
siteDepth: The depth of the URL space traversal in terms of authority hops. This is analogous to the pageDepth variable, but at a domain level. A server domain hop is made when a link points to a URL from a different server domain. Whenever a link is followed by the index server during the crawl to a different host (or item repository server), the site depth counter is incremented. The site depth counter cannot exceed the siteDepth of the content source.

startAddresses: A list of start address URLs. The first step of starting the crawl is to add the start address URLs to the crawl queue. The crawl then begins by following links from these start addresses.

fullCrawlTrigger: Defines the full crawl schedule. The crawl can be started either by explicit request from the protocol client, or automatically, at specified points of time, according to the schedule.

incCrawlTrigger: Defines the incremental crawl schedule.

crawlStatus: Identifies whether a crawl for this content source is idle, paused, stopped, or running. Also identifies what type of crawl it is, (full crawl or incremental crawl).

crawlStarted: The timestamp of when the most recent crawl was started for this content source.

crawlCompleted: The timestamp of when the most recent crawl was finished for this content source.

errorCount: The number of items, (during the most recent crawl), where the crawler attempted to crawl the items, but did not succeed.

crawlSuccesses: The number of items successfully crawled during the most recent crawl.

throttleStart: This property is not interpreted by the protocol server. It can be set and retrieved by the protocol client.

throttleDuration: This property is not interpreted by the protocol server. It can be set and retrieved by the protocol client.

3.1.1.5 Crawl Rule

The portal content project contains an ordered collection of zero or more crawl rule objects. Crawl rules are used to selectively restrict the URL space of the crawl. When a new link is discovered during the crawl, only the portion of the link up to, but not including, the query component is checked against the crawl rules to determine if the item will be included or excluded from the crawl.

Crawl rules contain a wildcard expression that can be matched against the URL of the link. All characters in the link are matched exactly against the expression, except the wildcards. Only two wildcard characters are allowed: '*' and '?'. The '*' wildcard character matches 0 or more arbitrary characters. The '?' wildcard character matches any single character.

Because of the presence of the wildcards in the crawl rules, the evaluation order of the rules is important. The first rule found to match the link determines the crawl behavior for that link.

The crawl rule can also specify authentication parameters for accessing items that match certain URLs.

path: The wildcard expression of the crawl rule. This also serves as a unique identifier of the crawl rule in the collection.

type: If 0, the links matching the crawl rule are included in the crawl URL Space; otherwise, excluded.

authenticationType: The authentication type the index server will use for the matching URLs. The following authentication types MUST be supported by the server:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default access</td>
</tr>
</tbody>
</table>

[MS-ADMWS] - v20171212
Search Service Administration Web Service Protocol
Copyright © 2017 Microsoft Corporation
Release: December 12, 2017
<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Integrated authentication</td>
</tr>
<tr>
<td>2</td>
<td>Basic authentication</td>
</tr>
<tr>
<td>3</td>
<td>Authentication using certificates</td>
</tr>
<tr>
<td>4</td>
<td>Forms authentication</td>
</tr>
<tr>
<td>5</td>
<td>Cookie based authentication</td>
</tr>
</tbody>
</table>

Default access implies integrated authentication using credentials of the default **crawl account** for the crawler application.

The following examples of possible implementation of various authentication types by the index server are provided for information only. The details are specific to the protocol server implementation.

- Integrated authentication using the account credentials specified for the crawl rule.
- Basic authentication can be accomplished by passing the account and the password to the item repository in plaintext.
- Authentication using certificates can be accomplished by presenting to the item repository a valid client certificate by the index server.
- Forms authentication can be accomplished by issuing an HTTP GET or HTTP POST against a specified form URL with required fields, acquiring a cookie from the form server and presenting the cookie in subsequent HTTP requests.
- Cookie based authentication can be accomplished by presenting a persistent cookie in the HTTP requests to the item repository.

**accountName:** The user name for integrated authentication or basic authentication.

**password:** The account password for integrated authentication or basic authentication, or private authentication parameters for forms authentication.

**contentClass:** Arbitrary metadata for the crawl rule. The content of this property can be interpreted by the index server as additional context used during the crawl of a particular item repository. As far as this protocol is concerned, the value of this property can be modified and retrieved by the protocol client.

**suppressIndexing:** If true, content of the URLs matching the crawl rule are not crawled, even if the crawl rule type is set to 0, (include). The links discovered from the matching URLs can still be followed during the crawl. If false, content of the URLs matching the crawl rule are crawled.

**followComplexUrls:** If true, links that have query components are followed by the crawl. Otherwise, links that have query components are discarded, even if the crawl rule type is 0.

**crawlAsHttp:** If true, matching links with an HTTP: scheme are crawled using the HTTP protocol, regardless of the default protocol normally used to crawl the repository. Crawling with the HTTP protocol implies that the process relies on discovered links instead of getting the repository to provide the set of links that need to be crawled. If false, matching links are crawled using protocols appropriate for the repository.

**enabled:** If true, the crawl rule is enabled; otherwise, the crawl rule is disabled.

**pluggableSecurityTrimmerId:** The protocol server can interpret this property as an identifier of a component that can validate if a user issuing the search query has read permission on the item. Any
item corresponding to the link matching this crawl rule will be assigned the security trimmer identifier if the `pluggableSecurityTrimmerId` was set for the crawl rule.

**authUrl**: When `authenticationType` is 4, this is the URL used for forms authentication. Otherwise, this element MUST be ignored.

**authData**: When `authenticationType` is 4, this is the portion of the opaque authentication BLOB used for forms authentication that does not need to be stored securely. Otherwise, this element MUST be ignored.

**authPrivateData**: When `authenticationType` is 4, this is the portion of the opaque authentication BLOB used for forms authentication that needs to be stored securely. Otherwise, this element MUST be ignored.

**authMethod**: When `authenticationType` is 4, this is the HTTP method, (HTTP GET or HTTP POST), used for forms authentication. Otherwise, this element MUST be ignored.

**miscData**: When `authenticationType` is 4 or 5, this is a list of error page URLs used to identify authentication errors. Otherwise, this element MUST be ignored.

**cookieName**: When `authenticationType` is 5, this is the opaque cookie used for cookie based authentication. Otherwise, this element MUST be ignored.

**certificateName**: When `authenticationType` is 3, this is the X.509 certificate used for authentication. Otherwise, this element MUST be ignored.

### 3.1.1.6 Anchor Content Source

The portal content project contains a single instance of the anchor content source. The anchor content source represents the status of the anchor crawl.

**crawlStatus**: The status of the anchor crawl. `crawlStatus` specifies whether the anchor crawl is currently in progress or not.

**crawlStarted**: The timestamp of when the last anchor crawl started.

**crawlCompleted**: The timestamp of when the last anchor crawl finished.

### 3.1.1.7 Crawl Mapping

The portal content project contains a collection of zero or more crawl mapping objects. Each crawled item has two URLs: access URL and display URL. The index server uses the access URL to obtain the item from the item repository, and it uses the display URL as a URL of the item to store in the metadata index. The display URL is returned to the users if the URL is requested in the search query. During the crawl the access URL and display URL of every item is checked against the crawl mapping objects. The match occurs if any prefix of the URL that covers complete path segments as described in [RFC2396], section 3.3, is equal to the Source or Target property of the mapping. If multiple mappings match the URL, the mapping that matches the longest prefix is chosen. For example, `http://site/pathseg1/pathseg2/file.htm` matches `http://site`, or `http://site/pathseg1`, or `http://site/pathseg1/pathseg2`, but does not match `http://site/pathseg1/path`.

If the access URL matches the Source property of the mapping, the matching prefix will be replaced by the Target property to construct the display URL. The suffix of the URL will be preserved.

The crawl mappings collection does not allow mappings with duplicate Source or Target properties.

- **Source**: source URL prefix for access URLs.
- **Target**: target URL prefix for the display URLs.
3.1.1.8 User Profile Import Project

The user profile import project contains configuration used by the index server to perform a user profile import from the Active Directory or any other repository of directory data. The crawler application contains a single instance of the user profile import project.

defaultAccountName: The user name used for authentication of the index server during the user profile import.

defaultAccountPassword: The password for the default account.

Enabled: If true, user profile import is enabled; otherwise it is disabled.

3.1.1.9 User Profile Content Source

The user profile import project contains exactly two user profile content sources, (with names "PEOPLE_IMPORT" and "PEOPLE_DL_IMPORT"). Each user profile content source contains the same properties as a section 3.1.1.4, with a single start address, and in addition has four more properties, which are:

lastCrawlType: If 0, no user profile import has been performed on the user profile content source before. If 1, the last user profile import was a full user profile import. If 2, the last user profile import was an incremental user profile import.

crawlInProgressType: The current user profile import type. If 0, no user profile import is currently in process. If 1, the current user profile import is a full user profile import. If 2, the current user profile import is an incremental user profile import.

lastCrawlSeedStatus: The crawl error code for the start address URL from the most recent crawl if the start address was not crawled successfully, otherwise, the value of this property will be 0.

lastCrawlErrorDescription: The description for the lastCrawlSeedStatus.

3.1.1.10 User Profile Import Domain

The user profile import project contains one or more user profile import domains. The user profile import domains specify what domains from the directory service (DS) are imported during the user profile import. The interpretation of the domain depends on the DS. In the case of Active Directory, there can be multiple domains defined in the directory, and only some of them are needed for the user profile import.

Domain: The domain name of the user profile import domain.

accountName: The user name used by the index server for authentication during a user profile import for this domain.

accountPassword: The password for the account specified in accountName.

3.1.2 Timers

None.

3.1.3 Initialization

None.
### 3.1.4 Message Processing Events and Sequencing Rules

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActivateCrawlRule</td>
<td>Activates a crawl rule.</td>
</tr>
<tr>
<td>ActivateProfileImport</td>
<td>Activates a user profile import.</td>
</tr>
<tr>
<td>AddApp</td>
<td>Not used.</td>
</tr>
<tr>
<td>AddContentSource</td>
<td>Creates a new content source.</td>
</tr>
<tr>
<td>AddCrawlMapping</td>
<td>Creates a new crawl mapping.</td>
</tr>
<tr>
<td>AddCrawlRule</td>
<td>Creates a new crawl rule.</td>
</tr>
<tr>
<td>AddExtension</td>
<td>Adds a new file extension.</td>
</tr>
<tr>
<td>AddSitePath</td>
<td>Not used.</td>
</tr>
<tr>
<td>AddSiteRestriction</td>
<td>Not used.</td>
</tr>
<tr>
<td>BackupRegistry</td>
<td>Backs up the server configuration.</td>
</tr>
<tr>
<td>CatalogPauseCrawlRequest</td>
<td>Requests that the crawl be paused.</td>
</tr>
<tr>
<td>CatalogPauseStatus</td>
<td>Returns the pause status of a search catalog</td>
</tr>
<tr>
<td>CatalogResumeCrawlRequest</td>
<td>Requests that a crawl be resumed.</td>
</tr>
<tr>
<td>CleanUnreferencedSiteRestrictions</td>
<td>Removes unreferenced user profile import domains.</td>
</tr>
<tr>
<td>ClearExtensionList</td>
<td>Removes all file extensions for the portal content project.</td>
</tr>
<tr>
<td>ConfigurePropagationShare</td>
<td>Not used.</td>
</tr>
<tr>
<td>DefaultInstallationIndexLocation</td>
<td>Not used.</td>
</tr>
<tr>
<td>DismountApp</td>
<td>Dismounts the crawler application.</td>
</tr>
<tr>
<td>EditContentSource</td>
<td>Modifies a content source’s properties.</td>
</tr>
<tr>
<td>GetContentSource</td>
<td>Retrieves a content source’s properties.</td>
</tr>
<tr>
<td>GetContentSources</td>
<td>Retrieves all the content sources for a specified search catalog.</td>
</tr>
<tr>
<td>GetContentState</td>
<td>Retrieves the crawler application’s properties.</td>
</tr>
<tr>
<td>GetCrawlMappings</td>
<td>Retrieves the list of all the crawl mappings.</td>
</tr>
<tr>
<td>GetCrawlRuleList</td>
<td>Retrieves the list of all the crawl rules.</td>
</tr>
<tr>
<td>GetExtensionList</td>
<td>Retrieves the list of all the file extensions.</td>
</tr>
<tr>
<td>GetGathererApplications</td>
<td>Not used.</td>
</tr>
<tr>
<td>GetImportAccounts</td>
<td>Retrieves all the user names for the user profile import.</td>
</tr>
<tr>
<td>GetImportStatus</td>
<td>Retrieves the current status of the user profile import.</td>
</tr>
<tr>
<td>GetPropagationInternal</td>
<td>Not used.</td>
</tr>
<tr>
<td>GetSiteRestrictionList</td>
<td>Not used.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetVersion</td>
<td>Retrieves the crawler application’s current configuration version.</td>
</tr>
<tr>
<td>IndexSize</td>
<td>Retrieves the size of the full-text index catalog.</td>
</tr>
<tr>
<td>IPAddress</td>
<td>Not used.</td>
</tr>
<tr>
<td>IsCatalogPauseCompleted</td>
<td>Retrieves the status of the request to pause the crawl.</td>
</tr>
<tr>
<td>IsDeleteCrawlInProgress</td>
<td>Retrieves the status of any delete crawl requests.</td>
</tr>
<tr>
<td>IsExtensionIncludeList</td>
<td>Retrieves the property indicating whether the file extensions list is an inclusion list or an exclusion list.</td>
</tr>
<tr>
<td>IsMounted</td>
<td>Not used.</td>
</tr>
<tr>
<td>IsPropagationShareConfigured</td>
<td>Not used.</td>
</tr>
<tr>
<td>ListKnownLotusNotesDatabases</td>
<td>Retrieves the list of all Lotus Notes databases known to the protocol server.</td>
</tr>
<tr>
<td>MountApp</td>
<td>Not used.</td>
</tr>
<tr>
<td>PauseCrawl</td>
<td>Requests that the protocol server pause the crawl.</td>
</tr>
<tr>
<td>RefreshAnchorContentSource</td>
<td>Retrieves the current status of the anchor content source.</td>
</tr>
<tr>
<td>RefreshContentSource</td>
<td>Retrieves the current status of the specified content source.</td>
</tr>
<tr>
<td>RemoveAllGathererApplications</td>
<td>Not used.</td>
</tr>
<tr>
<td>RemoveApp</td>
<td>Removes the crawler application state from the protocol server.</td>
</tr>
<tr>
<td>RemoveContentSource</td>
<td>Removes a content source.</td>
</tr>
<tr>
<td>RemoveCrawlMapping</td>
<td>Removes a crawl mapping.</td>
</tr>
<tr>
<td>RemoveCrawlRule</td>
<td>Removes a crawl rule.</td>
</tr>
<tr>
<td>RemoveExtension</td>
<td>Removes a file extension.</td>
</tr>
<tr>
<td>RemoveQueryServer</td>
<td>Not used.</td>
</tr>
<tr>
<td>RemoveSitePath</td>
<td>Not used.</td>
</tr>
<tr>
<td>RemoveSiteRestriction</td>
<td>Not used.</td>
</tr>
<tr>
<td>ResetApp</td>
<td>Resets the crawler application.</td>
</tr>
<tr>
<td>ResumeCrawl</td>
<td>Requests that a crawl be resumed.</td>
</tr>
<tr>
<td>SearchDatabaseCleanup</td>
<td>Enables search database cleanup.</td>
</tr>
<tr>
<td>SetContentSourcesMetadata</td>
<td>Updates the metadata property for the portal content project.</td>
</tr>
<tr>
<td>SetCrawlRuleCredentials</td>
<td>Updates the account for a crawl rule.</td>
</tr>
<tr>
<td>SetCrawlRulePriority</td>
<td>Updates the priority of a crawl rule.</td>
</tr>
<tr>
<td>SetDefaultGatheringAccount</td>
<td>Updates the default crawl account.</td>
</tr>
<tr>
<td>SetDefaultImportAccount</td>
<td>Updates the default user profile import account.</td>
</tr>
<tr>
<td>SetImportAccount</td>
<td>Updates the user profile import account on a specified user profile import domain.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetIsExtensionIncludeList</td>
<td>Toggles the state of the file extensions list between an inclusion list and an exclusion list.</td>
</tr>
<tr>
<td>SetRetryLimit</td>
<td>Updates the retry limit property of the crawler application.</td>
</tr>
<tr>
<td>StartCrawl</td>
<td>Requests that the protocol server start a crawl.</td>
</tr>
<tr>
<td>StartFullImport</td>
<td>Requests that the protocol server start a full user profile import.</td>
</tr>
<tr>
<td>StartIncrementalImport</td>
<td>Request that the protocol server start an incremental user profile import.</td>
</tr>
<tr>
<td>StartRankingUpdate</td>
<td>Requests that the protocol server start an anchor crawl.</td>
</tr>
<tr>
<td>StopCrawl</td>
<td>Requests that the protocol server stop a crawl.</td>
</tr>
<tr>
<td>StopImport</td>
<td>Requests that the protocol server stop a user profile import.</td>
</tr>
<tr>
<td>SystemDrive</td>
<td>Not used.</td>
</tr>
<tr>
<td>TestCrawlRule</td>
<td>Checks whether a URL matches a specified crawl rule.</td>
</tr>
<tr>
<td>TestCrawlRules</td>
<td>Returns the first crawl rule that matches the specified URL.</td>
</tr>
<tr>
<td>UpdateCrawlRule</td>
<td>Updates a crawl rule’s properties.</td>
</tr>
<tr>
<td>ValidateApplicationPath&lt;2&gt;</td>
<td>Validates that a folder path is a viable location to host a full-text index catalog.</td>
</tr>
<tr>
<td>ValidatePath</td>
<td>Not used.</td>
</tr>
<tr>
<td>ValidateScheduleTrigger</td>
<td>Validates that a trigger as defined in [MS-TSCH], section 2.4.2.11, can be used to schedule a crawl.</td>
</tr>
</tbody>
</table>

### 3.1.4.1 ActivateCrawlRule

This operation is used to enable an existing crawl rule.

```xml
<wsdl:operation name="ActivateCrawlRule">
  <wsdl:input message="ActivateCrawlRuleSoapIn"/>
  <wsdl:output message="ActivateCrawlRuleSoapOut"/>
</wsdl:operation>
```

The protocol client sends an ActivateCrawlRuleSoapIn request message and the server responds with an ActivateCrawlRuleSoapOut response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- The protocol server then updates the current configuration version of the crawler application to the value of `versionIn+1`.
- The protocol server MUST normalize the `path` element from the request message as follows:
  - If the path is empty or contains only white-space characters, the protocol server MUST throw a SOAP fault message containing a `COMException` with the `HRESULT` field set to 0x80040d34.

---

[MS-ADMWS] - v20171212
Search Service Administration Web Service Protocol
Copyright © 2017 Microsoft Corporation
Release: December 12, 2017
- If the length of the path is greater than or equal to 2048 characters, the protocol server MUST throw a SOAP fault message containing COMException with the HRESULT field set to 0x80040d14.
- If the path is a UNC path expression, the protocol server MUST convert it to the URL with a "file" URL scheme as described in [RFC2396], section 3.1.
- If the path is not a UNC path expression, the protocol server MUST replace all ‘\’ characters with the ‘/’ character.
- If the URL scheme, as described in [RFC2396], section 3.1 is not specified in the path, the protocol server MUST prepend path with the "http" scheme.
- If prepending the path with the "http" scheme causes the length of the resulting path to become greater than or equal to 2048 characters, the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80010105.
- If the host name in the path contains a '@' character, a '#' character, a '|' character, or a white-space character, the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040d50.
- If the URL scheme as described in [RFC2396], section 3.1 in the path is "http" or "https", the protocol server MUST convert it to a URL-encoded string.
- If the crawl rule identified by normalized path does not exist, the protocol server MUST throw a SOAP fault message containing a DeletedConcurrencyException.
- The protocol server MUST set the enabled property of the crawl rule to true.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the ActivateCrawlRuleResult element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.1.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.1.1.1 ActivateCrawlRuleSoapIn

This is the request message for the ActivateCrawlRule operation.

The SOAP action value of the message is defined as:


The SOAP body contains an ActivateCrawlRule element.

3.1.4.1.1.2 ActivateCrawlRuleSoapOut

This is a response message for the ActivateCrawlRule operation.

The SOAP action value of the message is defined as:

The SOAP body contains an ActivateCrawlRuleResponse element.

### 3.1.4.1.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.1.2.1 ActivateCrawlRule

The ActivateCrawlRule element contains parameters for the ActivateCrawlRule operation.

```xml
<s:element name="ActivateCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="path" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**path:** A crawl rule path expression. MUST be present, and the length MUST be greater than 0, and less than 2048 characters.

The protocol client SHOULD obtain this value from one of the following methods:
- AddCrawlRule
- GetCrawlRuleList
- TestCrawlRules

Otherwise, there is no guarantee that the protocol server will be able to identify the crawl rule, because the path of the crawl rule can be normalized by the protocol server.

#### 3.1.4.1.2.2 ActivateCrawlRuleResponse

The ActivateCrawlRuleResponse element contains the result of ActivateCrawlRule operation.

```xml
<s:element name="ActivateCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ActivateCrawlRuleResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**ActivateCrawlRuleResult:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

### 3.1.4.2 ActivateProfileImport

This operation is used to enable or disable the user profile import project.

```xml
<wsdl:operation name="ActivateProfileImport">
  <wsdl:input message="ActivateProfileImportSoapIn"/>
  <wsdl:output message="ActivateProfileImportSoapOut"/>
</wsdl:operation>
```
The protocol client sends an `ActivateProfileImportSoapIn` request message and the protocol server responds with an `ActivateProfileImportSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.

- The protocol server then updates the current configuration version of the crawler application to the value of `versionIn+1`.

- The protocol server sets the enabled property of the user profile import project to the value of the `enable` element from the request message.

- On success, the protocol server MUST return the updated configuration version of the crawler application in the `ActivateProfileImportResult` element in the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.2.1.1 `ActivateProfileImportSoapIn`

This is the request message for `ActivateProfileImport` operation.

The SOAP action value of the message is defined as:

```xml
http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/ActivateProfileImport
```

The SOAP body contains an `ActivateProfileImport` element.

#### 3.1.4.2.1.2 `ActivateProfileImportSoapOut`

This message is a response message for `ActivateProfileImport` operation.

The SOAP action value of the message is defined as:

```xml
http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/ActivateProfileImport
```

The SOAP body contains an `ActivateProfileImportResponse` element.

### 3.1.4.2.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.2.2.1 `ActivateProfileImport`

The definition of the `ActivateProfileImport` element is as follows:

```xml
<xs:element name="ActivateProfileImport">
  <xs:complexType>
    
  </xs:complexType>
</xs:element>
```
versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

disable: If true, the user profile import project needs to be enabled, otherwise, disabled.

3.1.4.2.2.2 ActivateProfileImportResponse

The definition of the ActivateProfileImportResponse element is as follows:

```
<s:element name="ActivateProfileImportResponse"/>
<s:complexType>
  <s:sequence>
    <s:element name="ActivateProfileImportResult" type="s:int"/>
  </s:sequence>
</s:complexType>
```

ActivateProfileImportResult: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.3 AddApp

This operation is not used and MUST NOT be invoked.

3.1.4.4 AddContentSource

This operation is used to create a new content source in the crawler application’s portal content project.

```
<wsdl:operation name="AddContentSource">
  <wsdl:input message="AddContentSourceSoapIn"/>
  <wsdl:output message="AddContentSourceSoapOut"/>
</wsdl:operation>
```

The protocol client sends an AddContentSourceSoapIn request message and the protocol server responds with an AddContentSourceSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- The protocol server creates a new content source with the properties type and name set to the values of the corresponding elements in the request message.
- The id property of the newly created content source MUST be unique within the collection of content sources.
If the content source is created successfully, the protocol server MUST return the content source identifier in the **AddContentSourceResult** element in the response message.

On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.4.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.4.1.1 AddContentSourceSoapIn

This is the request message for **AddContentSource** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains an **AddContentSource** element.

#### 3.1.4.4.1.2 AddContentSourceSoapOut

This is a response message for **AddContentSource** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains an **AddContentSourceResponse** element.

### 3.1.4.4.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.4.2.1 AddContentSource

The definition of the **AddContentSource** element is as follows:

```
<xs:element name="AddContentSource">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="versionIn" type="s:int"/>
      <xs:element name="catalog" type="s:int"/>  
      <xs:element name="type" type="s:int"/>
      <xs:element name="name" type="s:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**catalog:** This element MUST contain the value 0.

**type:** The content source type. MUST be one of the following values:
### Value | Meaning
---|---
0 | Web sites
1 | Sites
2 | Lotus Notes database
3 | File shares
4 | Exchange public folders
5 | Custom
6 | Business Data Connectivity (BDC)

**name:** The content source name. This element MUST be present. The length of the content source name MUST be less than or equal to 255 characters. The specified name MUST be unique among the content source names existing in the portal content project.

#### 3.1.4.4.2.2 AddContentSourceResponse

The definition of the **AddContentSourceResponse** element is as follows:

```xml
<s:element name="AddContentSourceResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="AddContentSourceResult" type="s:int"/>
        </s:sequence>
    </s:complexType>
</s:element>
```

**AddContentSourceResult:** Identifier of the newly created content source. MUST be greater than 0.

#### 3.1.4.5 AddCrawlMapping

This operation creates a new crawl mapping for the portal content project.

```xml
<wsdl:operation name="AddCrawlMapping">
    <wsdl:input message="AddCrawlMappingSoapIn"/>
    <wsdl:output message="AddCrawlMappingSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **AddCrawlMappingSoapIn** request message and the protocol server responds with an **AddCrawlMappingSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.
- The protocol server then updates the current configuration version of the crawler application to the value of **versionIn+1**.
- The protocol server MUST normalize the source and target strings as follows:
  - Trim the leading and trailing spaces.
- Remove the final character if it is the back slash (\) character, or the forward slash (/) character.
- If the source and target strings become the same, (case insensitive), after the previous step, the protocol server MUST throw a SOAP fault message containing a COMException, with the HRESULT field set to 0x80040d40.
- If the URL scheme as described in [RFC2396], section 3.1 is not present, and string is not an UNC path, the protocol server MUST prepend the path with the "http" scheme.
- If the URL scheme, as described in [RFC2396], section 3.1 is "file", convert the string to a UNC path, removing "file" prefix from the string, and replacing every forward slash (/) character with the back slash (\) character.
- If the URL scheme, as described in [RFC2396], section 3.1 is "http" or "https", replace every back slash (\) character with the forward slash (/) character.
- When normalization fails, if the failure is because of string exceeding maximum length, the protocol server MUST throw a SOAP fault message containing COMException with theHRESULT field set to 0x80040d14. Otherwise, the protocol server MUST throw a SOAP fault message containing COMException with the HRESULT field set to 0x80040d40.
- If the crawl mapping for the same normalized source or target exists, the protocol server MUST throw a SOAP fault message containing a COMException, with the HRESULT field set to 0x80040d02.
- The protocol server MUST then create a new crawl mapping and append it to the mappings collection in the portal content project with the source property set to the normalized source, and the target property set to the normalized target.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the AddCrawlMappingResult element in the response message.

On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.5.1.1 AddCrawlMappingSoapIn

This is the request message for AddCrawlMapping operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains an AddCrawlMapping element.

#### 3.1.4.5.1.2 AddCrawlMappingSoapOut

This is a response message for AddCrawlMapping operation.

The SOAP action value of the message is defined as:

```
```
The SOAP body contains an AddCrawlMappingResponse element.

### 3.1.4.5.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.5.2.1 AddCrawlMapping

The AddCrawlMapping element represents parameters for the AddCrawlMapping operation.

```xml
<s:element name="AddCrawlMapping">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="source" type="s:string" minOccurs="0"/>
      <s:element name="target" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**source:** The access URL prefix. This element MUST be present and its contents MUST be less than 2048 characters in length.

**target:** The display URL prefix. This element MUST be present and its contents MUST be less than 2048 characters in length.

#### 3.1.4.5.2.2 AddCrawlMappingResponse

The AddCrawlMappingResponse element represents the result for the AddCrawlMapping operation.

```xml
<s:element name="AddCrawlMappingResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddCrawlMappingResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**AddCrawlMappingResult:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

### 3.1.4.6 AddCrawlRule

This operation is used to create a new crawl rule for the portal content project.

```xml
<wsdl:operation name="AddCrawlRule">
  <wsdl:input message="AddCrawlRuleSoapIn"/>
  <wsdl:output message="AddCrawlRuleSoapOut"/>
</wsdl:operation>
```

The protocol client sends an AddCrawlRuleSoapIn request message and the protocol server responds with an AddCrawlRuleSoapOut response message, as follows:
On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.

The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.

The protocol server MUST normalize the path element of the request message the same way it normalizes the path element of the ActivateCrawlRule operation request message.

If a crawl rule with the same normalized path already exists in the portal content project, the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040d02.

If the protocol server has not thrown an exception, it MUST now create a new crawl rule with:

- path property set to the normalized path.
- type property set to the value of the isIncludeRule element from the request message
- enabled property set to true.

The protocol server MUST add the created crawl rule to the end of the crawl rules collection in the portal content project.

If the crawl rule was successfully added, the protocol server MUST return the normalized path property of the newly created crawl rule in the AddCrawlRuleResult element of the response message.

On error, the protocol server MUST throw a SOAP fault message.

3.1.4.6.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.6.1.1 AddCrawlRuleSoapIn

This is the request message for the AddCrawlRule operation.

The SOAP action value of the message is defined as:


The SOAP body contains an AddCrawlRule element.

3.1.4.6.1.2 AddCrawlRuleSoapOut

This is a response message for the AddCrawlRule operation.

The SOAP action value of the message is defined as:


The SOAP body contains an AddCrawlRuleResponse element.

3.1.4.6.2 Elements

The following XML Schema element definitions are specific to this operation.
3.1.4.6.2.1 AddCrawlRule

The **AddCrawlRule** element contains parameters for the **AddCrawlRule** operation.

```xml
<s:element name="AddCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="isIncludeRule" type="s:boolean"/>
      <s:element name="path" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**isIncludeRule:** An interpretation of the rule which MUST contain one of the values specified in the following table:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Inclusion rule. URLs matching the <strong>path</strong> are included in the crawl.</td>
</tr>
<tr>
<td>1</td>
<td>Exclusion rule. URLs matching the <strong>path</strong> are not included in the crawl.</td>
</tr>
</tbody>
</table>

**path:** A crawl rule path expression. MUST be present and the length MUST be greater than 0 and less than 2048 characters.

3.1.4.6.2.2 AddCrawlRuleResponse

The **AddCrawlRuleResponse** element contains the result of the **AddCrawlRule** operation.

```xml
<s:element name="AddCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddCrawlRuleResult" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**AddCrawlRuleResult:** The normalized crawl rule path expression used by the protocol server. MUST be present and the length MUST be greater than 0 and less than 2048 characters.

3.1.4.7 AddExtension

This operation is used to add a file extension to the file extensions collection contained in the portal content project.

```xml
<wsdl:operation name="AddExtension">
  <wsdl:input message="AddExtensionSoapIn"/>
  <wsdl:output message="AddExtensionSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **AddExtensionSoapIn** request message and the protocol server responds with an **AddExtensionSoapOut** response message, as follows:
On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.

The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.

The protocol server first removes any leading or trailing white-space characters from the ext element of the request message.

The protocol server then removes the first character of the ext element if it is the ‘.’ character.

If the ext element becomes empty, contains any white-space character, or contains any of the invalid characters (‘.’, ‘/’, ‘?’,’*, ‘\’, ‘#’), the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040d45.

If the file extension specified by the ext element already exists in the list, the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040D02.

The protocol server adds the file extension specified by the ext element to the portal content project’s list of extensions.

On success, the protocol server MUST return the updated configuration version of the crawler application in the AddExtensionResult element in the response message.

On error, the protocol server MUST throw a SOAP fault message.

3.1.4.7.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.7.1.1 AddExtensionSoapIn

This message is the request message for AddExtension operation.

The SOAP action value of the message is defined as:


The SOAP body contains an AddExtension element.

3.1.4.7.1.2 AddExtensionSoapOut

This is a response message for AddExtension operation.

The SOAP action value of the message is defined as:


The SOAP body contains an AddExtensionResponse element.

3.1.4.7.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.7.2.1 AddExtension
The definition of the **AddExtension** element is as follows:

```xml
<s:element name="AddExtension">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="ext" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**ext**: A file extension. MUST be present, and the length MUST be greater than 0 and less than 16 characters.

### 3.1.4.7.2.2 AddExtensionResponse

The definition of the **AddExtensionResponse** element is as follows:

```xml
<s:element name="AddExtensionResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddExtensionResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**AddExtensionResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

### 3.1.4.8 AddSitePath

This operation is not used and MUST NOT be invoked.

### 3.1.4.9 AddSiteRestriction

This operation is not used and MUST NOT be invoked.

### 3.1.4.10 BackupRegistry

This operation is used to notify the protocol server to save the configuration of the crawler application, including all contained objects and collections defined in the **crawler application**.

```xml
<wsdl:operation name="BackupRegistry">
  <wsdl:input message="BackupRegistrySoapIn"/>
  <wsdl:output message="BackupRegistrySoapOut"/>
</wsdl:operation>
```

The protocol client sends a **BackupRegistrySoapIn** request message and the protocol server responds with a **BackupRegistrySoapOut** response message, as follows:

- The protocol server starts backing up the configuration of the crawler application to the **configuration database** right away or after a delay as specified in the **backupNow** element of the request message.
On error, the protocol server MUST throw a SOAP fault message.

3.1.4.10.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.10.1.1 BackupRegistrySoapIn

This is the request message for BackupRegistry operation.

The SOAP action value of the message is defined as:


The SOAP body contains a BackupRegistry element.

3.1.4.10.1.2 BackupRegistrySoapOut

This is a response message for BackupRegistry operation.

The SOAP action value of the message is defined as:


The SOAP body contains a BackupRegistryResponse element.

3.1.4.10.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.10.2.1 BackupRegistry

The definition of the BackupRegistry element is as follows:

```xml
<s:element name="BackupRegistry">
    <s:complexType>
        <s:sequence>
            <s:element name="backupNow" type="s:boolean"/>
        </s:sequence>
    </s:complexType>
</s:element>
```

backupNow: If true, the crawler application configuration MUST be backed up immediately; otherwise with a delay of 15 seconds.

3.1.4.10.2.2 BackupRegistryResponse

The definition of the BackupRegistryResponse element is as follows:

```xml
<s:element name="BackupRegistryResponse"/>
```
3.1.4.11 CatalogPauseCrawlRequest

This operation is used to pause any crawl activity on the index server, including relevance computation, and update the pauseReason property of the portal content project.

```
<wSDL:operation name="CatalogPauseCrawlRequest">
  <wSDL:input message="CatalogPauseCrawlRequestSoapIn"/>
  <wSDL:output message="CatalogPauseCrawlRequestSoapOut"/>
</wSDL:operation>
```

The protocol client sends a CatalogPauseCrawlRequestSoapIn request message and the protocol server responds with a CatalogPauseCrawlRequestSoapOut response message, as follows:

- If the reason specified by the reason element of the request message is not one of the allowed values, the protocol server SHOULD throw a SOAP fault message containing an InvalidEnumArgumentException.
- If the pause reason specified in the reason element of the request message is already in the pause reasons stored in the pauseReason property of the portal content project, the protocol server MUST return right away.
- The protocol server MUST pause all crawl activities and add the pause reason specified in the reason element of the request message to the set of current pause reasons stored in the pauseReason property.
- When a request comes in to start a new crawl, the protocol server MUST pause the new crawl as well if the pauseReason property of the portal content project contains any pause reason.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.11.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.11.1.1 CatalogPauseCrawlRequestSoapIn

This is the request message CatalogPauseCrawlRequest operation.

The SOAP action value of the message is defined as:


The SOAP body contains a CatalogPauseCrawlRequest element.

3.1.4.11.1.2 CatalogPauseCrawlRequestSoapOut

This is a response message for CatalogPauseCrawlRequest operation.

The SOAP action value of the message is defined as:


The SOAP body contains a CatalogPauseCrawlRequestResponse element.
3.1.4.11.2  Elements
The following XML Schema element definitions are specific to this operation.

3.1.4.11.2.1  CatalogPauseCrawlRequest
The definition of the CatalogPauseCrawlRequest element is as follows:

```xml
<s:element name="CatalogPauseCrawlRequest">
  <s:complexType>
    <s:sequence>
      <s:element name="reason" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**reason:** The reason for pausing crawls. SHOULD be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Pause crawl because of back-up/restore process.</td>
</tr>
<tr>
<td>8</td>
<td>Pause crawl because of query server initialization.</td>
</tr>
</tbody>
</table>

3.1.4.11.2.2  CatalogPauseCrawlRequestResponse
The definition of the CatalogPauseCrawlRequestResponse element is as follows:

```xml
<s:element name="CatalogPauseCrawlRequestResponse"/>
```

3.1.4.12  CatalogPauseStatus
This operation is used to retrieve the pauseReason property of the portal content project.

```xml
<wsdl:operation name="CatalogPauseStatus">
  <wsdl:input message="CatalogPauseStatusSoapIn"/>
  <wsdl:output message="CatalogPauseStatusSoapOut"/>
</wsdl:operation>
```

The protocol client sends a CatalogPauseStatusSoapIn request message and the protocol server responds with a CatalogPauseStatusSoapOut response message, as follows:

- On success, the protocol server MUST return the pauseReason property of the portal content project, in the CatalogPauseStatusResult element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.12.1  Messages
The following WSDL message definitions are specific to this operation.

3.1.4.12.1.1  CatalogPauseStatusSoapIn
This is the request message for `CatalogPauseStatus` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `CatalogPauseStatus` element.

### 3.1.4.12.1.2 CatalogPauseStatusSoapOut

This is a response message for `CatalogPauseStatus` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `CatalogPauseStatusResponse` element.

### 3.1.4.12.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.12.2.1 CatalogPauseStatus

The definition of the `CatalogPauseStatus` element is as follows:

```
<s:element name="CatalogPauseStatus">
  <s:complexType/>
</s:element>
```

#### 3.1.4.12.2.2 CatalogPauseStatusResponse

The definition of the `CatalogPauseStatusResponse` element is as follows:

```
<s:element name="CatalogPauseStatusResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="CatalogPauseStatusResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**CatalogPauseStatusResult**: The pause reasons for the crawler application. The possible value SHOULD be derived from the bit wise OR of one or more of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000004</td>
<td>Because of back-up/restore.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Because of query server initialization.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>Because of an unresponsive query server.</td>
</tr>
</tbody>
</table>

If no pause reason is currently set, the protocol server MUST return 0x00000000.
3.1.4.13 CatalogResumeCrawlRequest

This operation is used to update the pauseReason property of the portal content project and to resume any previously paused crawl activity if there is no more pause reason remaining.

```xml
<wsdl:operation name="CatalogResumeCrawlRequest">
  <wsdl:input message="CatalogResumeCrawlRequestSoapIn"/>
  <wsdl:output message="CatalogResumeCrawlRequestSoapOut"/>
</wsdl:operation>
```

The protocol client sends a CatalogResumeCrawlRequestSoapIn request message and the protocol server responds with a CatalogResumeCrawlRequestSoapOut response message, as follows:

- If the reason specified by the reason element of the request message is not one of the allowed values, the protocol server SHOULD throw a SOAP fault message containing an InvalidEnumArgumentException.
- If the pause reason specified in the reason element of the request message is not in the pause reasons stored in the pauseReason property of the portal content project, the protocol server MUST return right away.
- Otherwise, the protocol server MUST update pauseReason property by removing the pause reason specified in the reason element of the request message from the pause reasons stored in the pauseReason property of the portal content project.
- If the pauseReason property becomes empty, the protocol server MUST resume any paused crawl.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.13.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.13.1.1 CatalogResumeCrawlRequestSoapIn

This is the request message for CatalogResumeCrawlRequest operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a CatalogResumeCrawlRequest element.

3.1.4.13.1.2 CatalogResumeCrawlRequestSoapOut

This is a response message for CatalogResumeCrawlRequest operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a CatalogResumeCrawlRequestResponse element.
3.1.4.13.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.13.2.1 CatalogResumeCrawlRequest

The definition of the CatalogResumeCrawlRequest element is as follows:

```xml
<s:element name="CatalogResumeCrawlRequest">
  <s:complexType>
    <s:sequence>
      <s:element name="reason" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

reason: The reason for pausing crawls. SHOULD be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Pause crawl because of back-up/restore process.</td>
</tr>
<tr>
<td>8</td>
<td>Pause crawl because of query server initialization.</td>
</tr>
</tbody>
</table>

3.1.4.13.2.2 CatalogResumeCrawlRequestResponse

The definition of the CatalogResumeCrawlRequestResponse element is as follows:

```xml
<s:element name="CatalogResumeCrawlRequestResponse"/>
```

3.1.4.14 CleanUnreferencedSiteRestrictions

This operation is used to remove user profile import domains that are not present in the specified list of domains.

```xml
<wsdl:operation name="CleanUnreferencedSiteRestrictions">
  <wsdl:input message="CleanUnreferencedSiteRestrictionsSoapIn"/>
  <wsdl:output message="CleanUnreferencedSiteRestrictionsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a CleanUnreferencedSiteRestrictionsSoapIn request message and the protocol server responds with a CleanUnreferencedSiteRestrictionsSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- The protocol server removes all user profile import domains from the user profile import projects that are not listed in one of the elements of the domains array element of the request message.
• On success, the protocol server MUST return the updated configuration version of the crawler application in the **CleanUnreferencedSiteRestrictionsResult** element in the response message.

• On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.14.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.14.1.1 CleanUnreferencedSiteRestrictionsSoapIn

This is the request message for **CleanUnreferencedSiteRestrictions** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **CleanUnreferencedSiteRestrictions** element.

#### 3.1.4.14.1.2 CleanUnreferencedSiteRestrictionsSoapOut

This is a response message for **CleanUnreferencedSiteRestrictions** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **CleanUnreferencedSiteRestrictionsResponse** element.

### 3.1.4.14.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.14.2.1 CleanUnreferencedSiteRestrictions

The **CleanUnreferencedSiteRestrictions** element represents parameters for the **CleanUnreferencedSiteRestrictions** operation.

```
<s:element name="CleanUnreferencedSiteRestrictions">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="domains" type="tns:ArrayOfString" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**domains**: An array of zero or more domain names for the user profile import domains to be kept. This element MUST be present. Each element in the array represents a single name of a user profile import domain. If the **domains** element is empty, that is the array does not have any element, the protocol server MUST remove all import domains from the user profile import project.
3.1.4.14.2.2 CleanUnreferencedSiteRestrictionsResponse

The CleanUnreferencedSiteRestrictions element represents the result of CleanUnreferencedSiteRestrictions operation.

```
<s:element name="CleanUnreferencedSiteRestrictionsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="CleanUnreferencedSiteRestrictionsResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

CleanUnreferencedSiteRestrictionsResult: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

3.1.4.15 ClearExtensionList

This operation is used to empty the list of file extensions recognized by the index server.

```
<wsd1:operation name="ClearExtensionList">
  <wsl:input message="ClearExtensionListSoapIn"/>
  <wsl:output message="ClearExtensionListSoapOut"/>
</wsd1:operation>
```

The protocol client sends a ClearExtensionListSoapIn request message and the protocol server responds with a ClearExtensionListSoapOut response message, as follows:

- On receipt, the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- The protocol server clears the extension property of the portal content project.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the ClearExtensionListResult element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.15.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.15.1.1 ClearExtensionListSoapIn

This is the request message for ClearExtensionList operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a ClearExtensionList element.
3.1.4.15.1.2 ClearExtensionListSoapOut

This is a response message for ClearExtensionList operation.

The SOAP action value of the message is defined as:


The SOAP body contains a ClearExtensionListResponse element.

3.1.4.15.2 Elements

3.1.4.15.2.1 ClearExtensionList

The definition of the ClearExtensionList element is as follows:

```xml
<s:element name="ClearExtensionList">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

3.1.4.15.2.2 ClearExtensionListResponse

The definition of the ClearExtensionListResponse element is as follows:

```xml
<s:element name="ClearExtensionListResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ClearExtensionListResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

ClearExtensionListResult: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

3.1.4.16 ConfigurePropagationShare

This operation is not used and MUST NOT be invoked.

3.1.4.17 DefaultInstallationIndexLocation

This operation is not used and MUST NOT be invoked.
3.1.4.18  DismountApp

This operation is used to dismount the crawler application. Dismounting of the crawler application means removing all state associated with the crawler application from the volatile memory, while preserving the persistent state.

<wsd1:operation name="DismountApp">  
  <wsdl:input message="DismountAppSoapIn"/>  
  <wsdl:output message="DismountAppSoapOut"/>  
</wsdl:operation>

The protocol client sends a DismountAppSoapIn request message and the protocol server responds with a DismountAppSoapOut response message, as follows:

- If the applicationName element is not present in the request message or the crawler application with the name specified by the applicationName element does not exist, the protocol server MUST return an empty DismountAppResponse element.
- Otherwise, the protocol server removes the crawler application and all the objects it contains from the volatile memory. After that an empty DismountAppResponse element MUST be returned.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.18.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.18.1.1  DismountAppSoapIn

This is the request message for DismountApp operation.

The SOAP action value of the message is defined as:


The SOAP body contains a DismountApp element.

3.1.4.18.1.2  DismountAppSoapOut

This is a response message for DismountApp operation.

The SOAP action value of the message is defined as:


The SOAP body contains a DismountAppResponse element.

3.1.4.18.2  Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.18.2.1  DismountApp

The DismountApp element represents parameters for DismountApp operation.
applicationName: The crawler application name. This element MUST be present.

3.1.4.18.2.2 DismountAppResponse

The DismountAppResponse element represents the result of DismountApp operation.

3.1.4.19 EditContentSource

This operation is used to edit the content source properties in the crawler application’s portal content project.

The protocol client sends an EditContentSourceSoapIn request message and the protocol server responds with a EditContentSourceSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- If the content source with the identifier specified in the id element in the request message does not exist, the protocol server MUST throw a SOAP fault message containing a DeletedConcurrencyException.
- The protocol server MUST normalize each element of the startAddress array element from the request message as follows:
  - If the string is not a valid URI or UNC path, the protocol server MUST throw SOAP fault message containing UriFormatException.
  - Convert the scheme to lowercase.
  - Convert the host name to lowercase.
  - Remove the default and empty port numbers.
  - Remove all trailing ‘/’ characters.
  - If the string represents a local path, the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040db2.
If the scheme is http or https, convert to a URL escaped string.

If the scheme is a file or the string is a UNC path then if the character `*` is present in the string, the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040d60.

If the scheme is file, convert the string to a UNC path, and replace every `/` character with the `\` character.

If the scheme is other than file, replace every `\` character with the `/` character.

If there are any duplicate elements of the `startAddress` array element in the request message, the protocol server MUST throw a SOAP fault message containing an ArgumentException.

If values of any of the elements of the `startAddress` array elements in the request message already exist as start addresses in another content source, and the `wssCrawlStyle` element of the request message is not 1, the protocol server MUST throw a SOAP fault message containing an ArgumentException with the `ParamName` parameter set to the duplicate value.

If the `wssCrawlStyle` element in the request message is set to 0, but there is another content source with a `wssCrawlStyle` property of 1 that contains a start address with the same host name as the host name of one of the elements of the `startAddresses` array element in the request message, the protocol server MUST throw a SOAP fault message containing an ArgumentException with the `ParamName` parameter set to the value of the failing element of the `startAddresses` array element.

The protocol server updates the content source’s `name`, `wssCrawlStyle`, `metadata`, `followDirectories`, `startAddress`, `fullCrawlTrigger` and `incCrawlTrigger` properties to the values specified in the corresponding elements in the request message, the `pageDepth` property to the value specified in the `enumerationDepth` element in the request message, and the `siteDepth` property to the value specified in the `hostDepth` element in the request message.

On success, the protocol server MUST return the updated configuration version of the crawler application in the `EditContentSourceResult` element in the response message.

On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.19.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.19.1.1 EditContentSourceSoapIn

This is a request message for `EditContentSource` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains an `EditContentSource` element.

#### 3.1.4.19.1.2 EditContentSourceSoapOut

This is the response message for `EditContentSource` operation.

The SOAP action value of the message is defined as:
The SOAP body contains an EditContentSourceResponse element.

### 3.1.4.19.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.19.2.1 EditContentSource

The definition of the EditContentSource element is as follows:

```xml
<s:element name="EditContentSource">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="id" type="s:int"/>
      <s:element name="name" type="s:string" minOccurs="0"/>
      <s:element name="metadata" type="s:string" minOccurs="0"/>
      <s:element name="hostDepth" type="s:int"/>
      <s:element name="enumerationDepth" type="s:int"/>
      <s:element name="followDirectories" type="s:boolean"/>
      <s:element name="wssCrawlStyle" type="s:int"/>
      <s:element name="throttleStart" type="s:int"/>
      <s:element name="throttleDuration" type="s:int"/>
      <s:element name="startAddresses" type="tns:ArrayOfString" minOccurs="0"/>
      <s:element name="fullCrawlTrigger" type="s:base64Binary" minOccurs="0"/>
      <s:element name="incCrawlTrigger" type="s:base64Binary" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

- **versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

- **id**: The content source identifier. The identifier MUST be greater than 0.

- **name**: The content source name. This element MUST be present. Length of the content source name MUST be greater than 0 characters, and less than or equal to 255 characters. The specified name MUST be unique among the content source names existing in the portal content project.

- **metadata**: Arbitrary metadata associated by the protocol client with the content source. If present, the length MUST be less than 1024 characters. The value of this property is ignored by the protocol server, but can be interpreted by the protocol client to associate arbitrary metadata with the content source collection.

- **hostDepth**: The number of host hops allowed when crawling this content source. The value MUST be greater than or equal to 0. For unlimited host hops the number MUST be 2147483647.

- **enumerationDepth**: The maximum number of page hops allowed when crawling this content source. The number MUST be greater than or equal to 0. For unlimited page hops the number MUST be 2147483647.

- **followDirectories**: Determines how URLs are discovered during the crawl. If true, URLs are discovered through directory links; otherwise, URLs are discovered through links.

- **wssCrawlStyle**: The scope of the start addresses. This property only applies to start addresses with URLs pointing to sites.
MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>All parts of Web applications are crawled.</td>
</tr>
<tr>
<td>1</td>
<td>Only the specific sites pointed to by the start addresses are crawled – the crawl does not enumerate all sites within the Web application.</td>
</tr>
</tbody>
</table>

**throttleStart**: The protocol server updates the `throttleStart` property of the content source, but does not interpret its value. SHOULD be set to 0.

**throttleDuration**: The protocol server updates the `throttleDuration` property of the content source, but does not interpret its value. SHOULD be set to 0.

**startAddresses**: An array of zero or more start addresses. This element MUST be present. Each element of the `startAddresses` array element MUST contain a single URL or UNC path that corresponds to a start address.

**fullCrawlTrigger**: A trigger as defined in [MS-TSCH], section 2.4.2.11 Triggers for scheduling full crawls. If this element is present, the content of the element MUST be a Base64 encoded trigger structure. If this element is not present, the protocol server SHOULD NOT define a schedule for the full crawl.

**incCrawlTrigger**: A Trigger as defined in [MS-TSCH], section 2.4.2.11 Triggers for scheduling incremental crawls. If this element is present, the content of the element MUST be Base64 encoded trigger structure. If this element is not present, the protocol server SHOULD NOT define a schedule for the incremental crawl.

### 3.1.4.19.2.2 EditContentSourceResponse

The definition of the `EditContentSourceResponse` element is as follows:

```xml
<s:element name="EditContentSourceResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="EditContentSourceResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**EditContentSourceResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

### 3.1.4.20 GetContentSource

This operation is used to get the current user profile content source properties and last user profile import information from the crawler application’s user profile import project.

```xml
<wsdl:operation name="GetContentSource">
  <wsdl:input message="GetContentSourceSoapIn"/>
  <wsdl:output message="GetContentSourceSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `GetContentSourceSoapIn` request message and the protocol server responds with a `GetContentSourceSoapOut` response message, as follows:
• On receipt the protocol server validates the **versionIn** value from the request message. If it
doesn’t match the current configuration version of the crawler application, the protocol server
MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.

• The protocol server looks up a user profile content source by the name specified in the
**contentSourceName** element of the request message. If the user profile content source does not
exists, the protocol server MUST throw a SOAP fault message.

• On success, the protocol server return the settings and last user profile import information for the
user profile content source in the **GetContentSourceResult** element of the response message.

• On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.20.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.20.1.1 GetContentSourceSoapIn

This is the request message for **GetContentSource** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **GetContentSource** element.

#### 3.1.4.20.1.2 GetContentSourceSoapOut

This is a response message for **GetContentSource** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **GetContentSourceResponse** element.

### 3.1.4.20.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.20.2.1 GetContentSource

The definition of the **GetContentSource** element is as follows:

```xml
<s:element name="GetContentSource">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="catalog" type="s:int"/>
      <s:element name="contentSourceName" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

catalog: This element MUST contain the value 1.

ccontentSourceName: This element MUST be present and MUST contain either "PEOPLE_IMPORT" or "PEOPLE_DL_IMPORT".

3.1.4.20.2.2 GetContentSourceResponse

The definition of the GetContentSourceResponse element is as follows:

```xml
<s:element name="GetContentSourceResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetContentSourceResult" type="tns:ContentSourceInternal" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetContentSourceResult: This element MUST be present and MUST contain current user profile content source properties and last user profile import information, as specified in ContentSourceInternal.

3.1.4.21 GetContentSources

This operation is used to get information about all the content sources for the specified project of the crawler application.

```xml
<wSDL:operation name="GetContentSources">
  <wSDL:input message="GetContentSourcesSoapIn"/>
  <wSDL:output message="GetContentSourcesSoapOut"/>
</wSDL:operation>
```

The protocol client sends a GetContentSourcesSoapIn request message and the protocol server responds with a GetContentSourcesSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server MUST obtain the properties of every content source in the project specified by catalog element in the request message.
- If the properties were successfully returned, the protocol server MUST return the properties of all content sources the GetContentSourcesResult element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.21.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.21.1.1 GetContentSourcesSoapIn

This is the request message for GetContentSources operation.
The SOAP action value of the message is defined as:

http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/GetContentSources

The SOAP body contains a **GetContentSources** element.

### 3.1.4.21.1.2 GetContentSourcesSoapOut

This is a response message for **GetContentSources** operation.

The SOAP action value of the message is defined as:

http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/GetContentSources

The SOAP body contains a **GetContentSourcesResponse** element.

### 3.1.4.21.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.21.2.1 GetContentSources

The definition of the **GetContentSources** element is as follows:

```
<s:element name="GetContentSources">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="catalog" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**catalog**: This element MUST contain a type of the project. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Portal content project</td>
</tr>
<tr>
<td>1</td>
<td>User profile import project</td>
</tr>
</tbody>
</table>

#### 3.1.4.21.2.2 GetContentSourcesResponse

The definition of the **GetContentSourcesResponse** element is as follows:

```
<s:element name="GetContentSourcesResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetContentSourcesResult" type="tns:ContentSourcesInternal"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
GetContentSourcesResult: All content sources for the catalog are specified in the request.

3.1.4.21.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

3.1.4.21.3.1 ContentSourcesInternal

The definition of the ContentSourcesInternal complex type is as follows:

```xml
<xs:complexType name="ContentSourcesInternal">
  <xs:sequence>
    <xs:element name="contentSourcesInternal" type="tns:ArrayOfContentSourceInternal" minOccurs="0"/>
    <xs:element name="metadata" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

contentSourcesInternal: This element MUST be present and MUST contain properties of each content source.

metadata: Arbitrary metadata associated by the protocol client with the content source. If present, the length MUST be less than 1024 characters. The value of this property is ignored by the protocol server, but can be interpreted by the protocol client to associate arbitrary metadata with the content source collection.

3.1.4.21.3.2 ArrayOfContentSourceInternal

The ArrayOfContentSourceInternal complex type represents an array of elements of a size equal to the number of content sources present in the project. Each element MUST contain the properties for a content source.

The definition of the ArrayOfContentSourceInternal complex type is as follows:

```xml
<xs:complexType name="ArrayOfContentSourceInternal">
  <xs:sequence>
    <xs:element name="ContentSourceInternal" type="tns:ContentSourceInternal" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

ContentSourceInternal: This element MUST be present and MUST NOT be null, and MUST contain a single content source’s properties, according to the ContentSourceInternal common type.

3.1.4.22 GetContentState

This operation is used to retrieve the states and various properties of the crawler application and the portal content project.
The protocol client sends a `GetContentStateSoapIn` request message and the protocol server responds with a `GetContentStateSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.

- On success, the protocol server MUST return the states and various properties of the crawler application and the portal content project in the `GetContentStateResult` element of the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.22.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.22.1.1 GetContentStateSoapIn

This is the request message for `GetContentState` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `GetContentState` element.

#### 3.1.4.22.1.2 GetContentStateSoapOut

This is a response message for `GetContentState` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `GetContentStateResponse` element.

### 3.1.4.22.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.22.2.1 GetContentState

The definition of the `GetContentState` element is as follows:

```xml
<s:element name="GetContentState">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

### 3.1.4.22.2 GetContentStateResponse

The definition of the **GetContentStateResponse** element is as follows:

```xml
<s:element name="GetContentStateResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetContentStateResult" type="tns:ContentStateInternal" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetContentStateResult:** Properties of the crawler application and the portal content project. This element MUST be present and MUST conform to the schema of the **ContentStateInternal** complex type.

### 3.1.4.22.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

#### 3.1.4.22.3.1 ContentStateInternal

The definition of the **ContentStateInternal** type is as follows:

```xml
<s:complexType name="ContentStateInternal">
  <s:sequence>
    <s:element name="retryLimit" type="s:int"/>
    <s:element name="lotusNotesConfigured" type="s:boolean"/>
    <s:element name="defaultGatheringAccount" type="s:string" minOccurs="0"/>
    <s:element name="clientCertificateNames" type="tns:ArrayOfString" minOccurs="0"/>
    <s:element name="activeDocuments" type="tns:ArrayOfString" minOccurs="0"/>
    <s:element name="listKnownLotusNotesServers" type="tns:ArrayOfString" minOccurs="0"/>
  </s:sequence>
</s:complexType>
```

The purpose of the following properties is detailed in the Abstract Data Model (section 3.1.1):

- **retryLimit:** The **retryLimit** property of the crawler application.
- **lotusNotesConfigured:** The **lotusNotesConfigured** property of the crawler application.
- **defaultGatheringAccount:** The **defaultCrawlingAccount** property of the crawler application. This element MUST be present.
- **clientCertificateNames:** The **clientCertificateNames** property of the portal content project. This element MUST be present and MUST conform to the schema of the **ArrayOfString** complex type.
- **activeDocuments:** The **activeDocuments** property of the portal content project. This element MUST be present and MUST conform to the schema of the **ArrayOfString** complex type.
- **listKnownLotusNotesServers:** The **listKnownLotusNotesServers** property of the crawler application. If the **lotusNotesConfigured** property of the crawler application is true, this element MUST be present and MUST conform to the schema of the **ArrayOfString** complex type.
3.1.4.23  GetCrawlMappings

This method is used to retrieve all crawl mappings that exist in the portal content project.

```
<wsdl:operation name="GetCrawlMappings">
  <wsdl:input message="GetCrawlMappingsSoapIn"/>
  <wsdl:output message="GetCrawlMappingsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a GetCrawlMappingsSoapIn request message and the protocol server responds with a GetCrawlMappingsSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- On success, the protocol server MUST return all crawl mappings existing in the portal content project in the GetCrawlMappingsResult element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.23.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.23.1.1  GetCrawlMappingsSoapIn

This is the request message for GetCrawlMappings (section 3.1.4.23) operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a GetCrawlMappings element.

3.1.4.23.1.2  GetCrawlMappingsSoapOut

This is a response message for GetCrawlMappings operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a GetCrawlMappingsResponse element.

3.1.4.23.2  Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.23.2.1  GetCrawlMappings

The GetCrawlMappings element represents the parameter for the GetCrawlMappings operation.

```
<s:element name="GetCrawlMappings">
```
<s:complexType>
  <s:sequence>
    <s:element name="versionIn" type="s:int"/>
  </s:sequence>
</s:complexType>

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

### 3.1.4.23.23.2.2 GetCrawlMappingsResponse

The **GetCrawlMappingsResponse** element represents results for the **GetCrawlMappings** operation.

GetCrawlMappingsResult: An array of crawl mappings. This element MUST be present.

### 3.1.4.23.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

### 3.1.4.23.3.1 ArrayOfCrawlMappingInternal

The **ArrayOfCrawlMappingInternal** complex type represents an array of zero or more elements, each containing a single crawl mapping. Each crawl mapping MUST be presented once.

CrawlMappingInternal: A single crawl mapping. If present, this element MUST NOT be null.

### 3.1.4.23.3.2 CrawlMappingInternal

This element represents a crawl mapping.

**source**: An access URL prefix. This element MUST be present and its content MUST be less than 2048 characters in length.
target: A display URL prefix. This element MUST be present and its content MUST be less than 2048 characters in length.

3.1.4.24  GetCrawlRuleList

This operation is used to retrieve the portal content project’s list of crawl rules.

```
<wsl:operation name="GetCrawlRuleList">
  <wsl:input message="GetCrawlRuleListSoapIn"/>
  <wsl:output message="GetCrawlRuleListSoapOut"/>
</wsl:operation>
```

The protocol client sends a GetCrawlRuleListSoapIn request message and the protocol server responds with a GetCrawlRuleListSoapOut response message, as follows:

On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.

- If the operation succeeds, the protocol server MUST return all crawl rules in the GetCrawlRuleListResult element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.24.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.24.1.1  GetCrawlRuleListSoapIn

This is the request message for the GetCrawlRuleList operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a GetCrawlRuleList element.

3.1.4.24.1.2  GetCrawlRuleListSoapOut

This is a response message for the GetCrawlRuleList operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a GetCrawlRuleListResponse element.

3.1.4.24.2  Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.24.2.1  GetCrawlRuleList
This element represents parameters for GetCrawlRuleList operation.

```xml
<s:element name="GetCrawlRuleList">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

### 3.1.4.24.2.2 GetCrawlRuleListResponse

This element represents the result of GetCrawlRuleList operation.

```xml
<s:element name="GetCrawlRuleListResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetCrawlRuleListResult" type="tns:ArrayOfCrawlRuleInternal" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetCrawlRuleListResult**: The list of crawl rules. This element MUST be present if there is at least one crawl rule.

### 3.1.4.24.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

#### 3.1.4.24.3.1 ArrayOfCrawlRuleInternal

This complex type represents an ordered list of crawl rules with the ordering of the CrawlRuleInternal elements the same as the ordering of the crawl rules in the portal content project.

```xml
<s:complexType name="ArrayOfCrawlRuleInternal">
  <s:sequence>
    <s:element name="CrawlRuleInternal" type="tns:CrawlRuleInternal" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

**CrawlRuleInternal**: The properties of a crawl rule. If present, this element MUST NOT be null.

### 3.1.4.25 GetExtensionList

This operation is used to retrieve the portal content project’s list of file extensions.
The protocol client sends a `GetExtensionListSoapIn` request message and the protocol server responds with a `GetExtensionListSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.

- The protocol server returns all file extensions from the portal content project’s extension list in the `GetExtensionListResult` element in the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.25.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.25.1.1 GetExtensionListSoapIn

This is the request message for `GetExtensionList` operation

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `GetExtensionList` element.

#### 3.1.4.25.1.2 GetExtensionListSoapOut

This is a response message for `GetExtensionList` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `GetExtensionListResponse` element.

### 3.1.4.25.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.25.2.1 GetExtensionList

The definition of the `GetExtensionList` element is as follows:

```xml
<s:element name="GetExtensionList">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

`versionIn`: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.
3.1.4.25.2.2 GetExtensionListResponse

The definition of the GetExtensionListResponse element is as follows:

```xml
<xs:element name="GetExtensionListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="GetExtensionListResult" type="tns:ArrayOfString" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**GetExtensionListResult**: An array of zero or more file extensions that the protocol server returns to the protocol client. This element MUST be present. Each element of the GetExtensionListResult array element MUST NOT be null and MUST contain one file extension.

3.1.4.26 GetGathererApplications

This operation is not used and MUST NOT be invoked.

3.1.4.27 GetImportAccounts

This operation is used to retrieve the default user profile import account and the user profile import accounts for all user profile import domains.

```xml
<wsdl:operation name="GetImportAccounts">
  <wsdl:input message="GetImportAccountsSoapIn"/>
  <wsdl:output message="GetImportAccountsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a GetImportAccountsSoapIn request message and the protocol server responds with a GetImportAccountsSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.

- On success, the protocol server returns, in the GetImportAccountsResult element of the response message, a list of user profile import accounts for all user profile import domains along with three special accounts in the first, second, and last positions of the list. The first entry of the list MUST contain an account element with its value set to the default content access account and an empty domain element. The second entry of the list MUST contain an account element with its value set to the default user profile import account and an empty domain element. The last entry of the list MUST contain a domain element with its value set to the identifier of the SSP hosting the crawler application and an account element with its value set to the default user profile import account.

- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.27.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.27.1.1 GetImportAccountsSoapIn

This is the request message for GetImportAccounts operation.
The SOAP action value of the message is defined as:


The SOAP body contains a GetImportAccounts element.

3.1.4.27.1.2 GetImportAccountsSoapOut

This is a response message for GetImportAccounts operation.

The SOAP action value of the message is defined as:


The SOAP body contains a GetImportAccountsResponse element.

3.1.4.27.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.27.2.1 GetImportAccounts

The definition of the GetImportAccounts element is as follows:

```xml
<s:element name="GetImportAccounts">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

3.1.4.27.2.2 GetImportAccountsResponse

The definition of the GetImportAccountsResponse element is as follows:

```xml
<s:element name="GetImportAccountsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetImportAccountsResult" type="tns:ArrayOfImportDomainAccount" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetImportAccountsResult: The user profile import accounts. This element MUST be present and MUST conform to the schema of the ArrayOfImportDomainAccount complex type.

3.1.4.27.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.
3.1.4.27.3.1 ArrayOfImportDomainAccount
The definition of the **ArrayOfImportDomainAccount** type is as follows:

```xml
<s:complexType name="ArrayOfImportDomainAccount">
    <s:sequence>
        <s:element name="ImportDomainAccount" type="tns:ImportDomainAccount" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
    </s:sequence>
</s:complexType>
```

**ImportDomainAccount**: A user profile import account. If present, this element MUST conform to the schema of the **ImportDomainAccount** complex type.

3.1.4.27.3.2 ImportDomainAccount
The definition of the **ImportDomainAccount** type is as follows:

```xml
<s:complexType name="ImportDomainAccount">
    <s:sequence>
        <s:element name="domain" type="s:string" minOccurs="0"/>
        <s:element name="account" type="s:string" minOccurs="0"/>
    </s:sequence>
</s:complexType>
```

domain: The domain name of an user profile import domain. This element MUST be present.

account: The user profile import account for the user profile import domain. This element MUST be present.

3.1.4.28 GetImportStatus
This operation is used to retrieve the user profile import status of a user profile content source.

```xml
<wSDL:operation name="GetImportStatus">
    <wSDL:input message="GetImportStatusSoapIn"/>
    <wSDL:output message="GetImportStatusSoapOut"/>
</wSDL:operation>
```

The protocol client sends a **GetImportStatusSoapIn** request message and the protocol server responds with a **GetImportStatusSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.

- On success, the protocol server returns, in the **GetImportStatusResult** element of the response message, the user profile import status of the user profile content source specified by the **contentSourceName** element of the request message.

- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.28.1 Messages
The following WSDL message definitions are specific to this operation.

3.1.4.28.1.1 GetImportStatusSoapIn
This is the request message for GetImportStatus operation. The SOAP action value of the message is defined as:


The SOAP body contains a GetImportStatus element.

### 3.1.4.28.1.2 GetImportStatusSoapOut

This is a response message for GetImportStatus operation. The SOAP action value of the message is defined as:


The SOAP body contains a GetImportStatusResponse element.

### 3.1.4.28.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.28.2.1 GetImportStatus

The definition of the GetImportStatus element is as follows:

```xml
<s:element name="GetImportStatus">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="contentSourceName" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**contentSourceName**: The user profile content source name for which to get the status. This element MUST be present. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEOPLE_IMPORT</td>
<td>The user profile content source name.</td>
</tr>
<tr>
<td>PEOPLE_DL_IMPORT</td>
<td>The distribution list import content source name.</td>
</tr>
</tbody>
</table>

#### 3.1.4.28.2.2 GetImportStatusResponse

The definition of the GetImportStatusResponse element is as follows:

```xml
<s:element name="GetImportStatusResponse">
```
GetImportStatusResult: The user profile import status of a user profile content source. This element MUST be present and MUST conform to the schema of the StartAddressDynamicPropsInternal complex type.

3.1.4.28.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

3.1.4.28.3.1 StartAddressDynamicPropsInternal

The definition of the StartAddressDynamicPropsInternal type is as follows:

```
<s:complexType name="StartAddressDynamicPropsInternal">
  <s:sequence>
    <s:element name="lastCrawlType" type="s:int"/>
    <s:element name="crawlInProgressType" type="s:int"/>
    <s:element name="lastCrawlSeedStatus" type="s:int"/>
    <s:element name="lastCrawlErrorDescription" type="s:string" minOccurs="0"/>
  </s:sequence>
</s:complexType>
```

lastCrawlType: The type of the last user profile import. If 0, no user profile import has been performed on the user profile content source. If 1, the last user profile import was a full user profile import. If 2, the last user profile import was an incremental user profile import.

crawlInProgressType: The type of current user profile import. If 0, there is no profile import in progress. If 1, the current user profile import is a full user profile import. If 2, the current user profile import is an incremental user profile import.

lastCrawlSeedStatus: The crawl error code for the start address URL from the most recent crawl if the start address was not crawled successfully; otherwise, the value of this property will be 0.

lastCrawlErrorDescription: If present, the text description of the lastCrawlSeedStatus.

3.1.4.29 GetPropagationInternal

This operation is not used and MUST NOT be invoked.

3.1.4.30 GetSiteRestrictionList

This operation is not used and MUST NOT be invoked.

3.1.4.31 GetVersion

This operation is used to retrieve the configuration version of the crawler application.
The protocol client sends a `GetVersionSoapIn` request message and the protocol server responds with a `GetVersionSoapOut` response message, as follows:

- On success, the protocol server MUST return the configuration version property of the crawler application in the `GetVersionResult` element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.31.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.31.1.1 `GetVersionSoapIn`

This is the request message for the `GetVersion` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `GetVersion` element.

#### 3.1.4.31.1.2 `GetVersionSoapOut`

This is a response message for the `GetVersion` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `GetVersionResponse` element.

### 3.1.4.31.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.31.2.1 `GetVersion`

The definition of the `GetVersion` element is as follows:

```
<xs:element name="GetVersion">
  <xs:complexType/>
</xs:element>
```

#### 3.1.4.31.2.2 `GetVersionResponse`

The definition of the `GetVersionResponse` element is as follows:

```
<xs:element name="GetVersionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="GetVersionResult" type="s:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
GetVersionResult: The updated configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

3.1.4.32  IndexSize

This operation is used to retrieve the size of the crawler application’s full-text index catalog directory on disk for the host SSP of this operation.

```xml
<wsdl:operation name="IndexSize">
  <wsdl:input message="IndexSizeSoapIn"/>
  <wsdl:output message="IndexSizeSoapOut"/>
</wsdl:operation>
```

The protocol client sends a IndexSizeSoapIn request message and the protocol server responds with a IndexSizeSoapOut response message, as follows:

- If the full-text index catalog does not exist, the protocol server MUST return zero in the IndexSizeResult element of the response message.
- On success, the protocol server returns the size of the full-text index catalog, in bytes, in the IndexSizeResult element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.32.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.32.1.1  IndexSizeSoapIn

This is the request message for IndexSize operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains an IndexSize element.

3.1.4.32.1.2  IndexSizeSoapOut

This is a response message for IndexSize operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains an IndexSizeResponse element.

3.1.4.32.2  Elements

The following XML Schema element definitions are specific to this operation.
3.1.4.32.2.1 IndexSize

The definition of the **IndexSize** element is as follows:

```xml
<s:element name="IndexSize">
  <s:complexType/>
</s:element>
```

3.1.4.32.2.2 IndexSizeResponse

The definition of the **IndexSizeResponse** element is as follows:

```xml
<s:element name="IndexSizeResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="IndexSizeResult" type="s:long"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**IndexSizeResult:** The size of the full-text index catalog in bytes.

3.1.433 IPAddress

This operation is not used and MUST NOT be invoked.

3.1.434 IsCatalogPauseCompleted

This operation is used to check if the action of pausing all crawls on the portal content project for the specified reason has been finished.

```xml
<wSDL:operation name="IsCatalogPauseCompleted">
  <wSDL:input message="IsCatalogPauseCompletedSoapIn"/>
  <wSDL:output message="IsCatalogPauseCompletedSoapOut"/>
</wSDL:operation>
```

The protocol client sends an **IsCatalogPauseCompletedSoapIn** request message and the protocol server responds with an **IsCatalogPauseCompletedSoapOut** response message, as follows:

- If the value of the crawl pause reason is successfully retrieved, the protocol server MUST return whether the **pauseReason** property on the portal content project contains the reason specified in the reason element of the request message in the **IsCatalogPauseCompletedResult** element of the response message.

- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.34.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.34.1.1 IsCatalogPauseCompletedSoapIn

This is the request message for **IsCatalogPauseCompleted** operation.

The SOAP action value of the message is defined as:
The SOAP body contains an IsCatalogPauseCompleted element.

**3.1.4.34.1.2 IsCatalogPauseCompletedSoapOut**

This is a response message for IsCatalogPauseCompleted operation.

The SOAP action value of the message is defined as:


The SOAP body contains an IsCatalogPauseCompletedResponse element.

**3.1.4.34.2 Elements**

The following XML Schema element definitions are specific to this operation.

**3.1.4.34.2.1 IsCatalogPauseCompleted**

The definition of the IsCatalogPauseCompleted element is as follows:

```
<s:element name="IsCatalogPauseCompleted">
  <s:complexType>
    <s:sequence>
      <s:element name="reason" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**reason:** The reason for pausing all crawls. SHOULD be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Because of back-up/restore.</td>
</tr>
<tr>
<td>8</td>
<td>Because of query server initialization.</td>
</tr>
</tbody>
</table>

**3.1.4.34.2.2 IsCatalogPauseCompletedResponse**

The definition of the IsCatalogPauseCompletedResponse element is as follows:

```
<s:element name="IsCatalogPauseCompletedResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="IsCatalogPauseCompletedResult" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**IsCatalogPauseCompletedResult:** If true, indicates the operation to pause all crawls for the given reason has been finished; otherwise false.
3.1.4.35  IsDeleteCrawlInProgress

This operation is used to check if a delete crawl in the crawler application is in progress.

```xml
<wsdl:operation name="IsDeleteCrawlInProgress">
  <wsdl:input message="IsDeleteCrawlInProgressSoapIn"/>
  <wsdl:output message="IsDeleteCrawlInProgressSoapOut"/>
</wsdl:operation>
```

The protocol client sends an `IsDeleteCrawlInProgressSoapIn` request message and the protocol server responds with a `IsDeleteCrawlInProgressSoapOut` response message, as follows:

- On success, the protocol server MUST return whether a delete crawl is in progress in the `IsDeleteCrawlInProgressResult` element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.35.1  Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.35.1.1  IsDeleteCrawlInProgressSoapIn

This is the request message for `IsDeleteCrawlInProgress` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains an `IsDeleteCrawlInProgress` element.

#### 3.1.4.35.1.2  IsDeleteCrawlInProgressSoapOut

This is a response message for `IsDeleteCrawlInProgress` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains an `IsDeleteCrawlInProgressResponse` element.

### 3.1.4.35.2  Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.35.2.1  IsDeleteCrawlInProgress

The definition of the `IsDeleteCrawlInProgress` element is as follows:

```xml
<s:element name="IsDeleteCrawlInProgress">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.35.2.2 IsDeleteCrawlInProgressResponse

The definition of the IsDeleteCrawlInProgressResponse element is as follows:

```
<s:element name="IsDeleteCrawlInProgressResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="IsDeleteCrawlInProgressResult" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

IsDeleteCrawlInProgressResult: If true, indicates that a delete crawl is in progress, otherwise false.

3.1.4.36 IsExtensionIncludeList

This operation is used to determine whether the file extensions list in the portal content project is an inclusion list or an exclusion list.

The protocol client sends an IsExtensionIncludeListSoapIn request message and the protocol server responds with an IsExtensionIncludeListSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server returns the IsExtensionIncludeListResult property of the portal content project in the IsExtensionIncludeListResult element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.36.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.36.1.1 IsExtensionIncludeListSoapIn

This is the request message for IsExtensionIncludeList operation.

The SOAP action value of the message is defined as:

The SOAP body contains an IsExtensionIncludeList element.

### 3.1.4.36.1.2 IsExtensionIncludeListSoapOut

This is a response message for IsExtensionIncludeList operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains an IsExtensionIncludeListResponse element.

### 3.1.4.36.2 Elements

#### 3.1.4.36.2.1 IsExtensionIncludeList

The definition of the IsExtensionIncludeList element is as follows:

```
<s:element name="IsExtensionIncludeList">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

#### 3.1.4.36.2.2 IsExtensionIncludeListResponse

The definition of the IsExtensionIncludeListResponse element is as follows:

```
<s:element name="IsExtensionIncludeListResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="IsExtensionIncludeListResult" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**IsExtensionIncludeListResult**: The type of the file extensions list. If true, the file extensions list is an exclusion list, otherwise, the file extensions list is an inclusion list.

### 3.1.4.37 IsMounted

This operation is not used and MUST NOT be invoked.

### 3.1.4.38 IsPropagationShareConfigured

This operation is not used and MUST NOT be invoked.
3.1.4.39 ListKnownLotusNotesDatabases

This operation is used to retrieve a list of known Lotus Notes database names for a given Lotus Notes server name.

```xml
<wsdl:operation name="ListKnownLotusNotesDatabases">
  <wsdl:input message="ListKnownLotusNotesDatabasesSoapIn"/>
  <wsdl:output message="ListKnownLotusNotesDatabasesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `ListKnownLotusNotesDatabasesSoapIn` request message and the protocol server responds with a `ListKnownLotusNotesDatabasesSoapOut` response message, as follows:

- If the `lotusNotesConfigured` property of the crawler application is not true, the protocol server MUST throw SOAP fault message containing `InvalidOperationException`.
- On success, the protocol server returns, in the `ListKnownLotusNotesDatabasesResult` element the response message, the known Lotus Notes database names for the Lotus Notes server name specified in the `serverName` element of the request message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.39.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.39.1.1 ListKnownLotusNotesDatabasesSoapIn

This is the request message for `ListKnownLotusNotesDatabases` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ListKnownLotusNotesDatabases` element.

3.1.4.39.1.2 ListKnownLotusNotesDatabasesSoapOut

This is a response message for `ListKnownLotusNotesDatabases` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ListKnownLotusNotesDatabasesResponse` element.

3.1.4.39.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.39.2.1 ListKnownLotusNotesDatabases

The definition of the `ListKnownLotusNotesDatabases` element is as follows:
<s:element name="ListKnownLotusNotesDatabases">
  <s:complexType>
    <s:sequence>
      <s:element name="serverName" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>

**serverName**: Lotus Notes server name for which the Lotus Notes database names MUST be retrieved. This element MUST be present and MUST NOT be empty.

### 3.1.4.39.2.2 ListKnownLotusNotesDatabasesResponse

The definition of the `ListKnownLotusNotesDatabasesResponse` element is as follows:

<ts:element name="ListKnownLotusNotesDatabasesResponse">
  <ts:complexType>
    <ts:sequence>
      <ts:element name="ListKnownLotusNotesDatabasesResult" type="tns:ArrayOfString" minOccurs="0"/>
    </ts:sequence>
  </ts:complexType>
</ts:element>

**ListKnownLotusNotesDatabasesResult**: A list of Lotus Notes database names. This element MUST be present and MUST conform to the schema of `ArrayOfString` complex type.

### 3.1.4.40 MountApp

This operation is not used and MUST NOT be invoked.

### 3.1.4.41 PauseCrawl

This operation is used to pause a crawl of a content source of the portal content project.

<wsdl:operation name="PauseCrawl">
  <wsdl:input message="PauseCrawlSoapIn"/>
  <wsdl:output message="PauseCrawlSoapOut"/>
</wsdl:operation>

The protocol client sends a `PauseCrawlSoapIn` request message and the protocol server responds with a `PauseCrawlSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- If there is no content source with the content source identifier specified by the `contentSource` element of the request message, the protocol server MUST throw a SOAP fault message containing a `DeletedConcurrencyException`.
- The protocol server MUST try to pause the crawl for the content source with the identifier specified in the `contentSource` element of the request message.
- If the crawl is in a state where it cannot be paused or is already paused, the protocol server MUST ignore the pause request and return the current crawl status in the `PauseCrawlResult` element the response message.
• On success, the protocol server MUST return the current crawl status in the **PauseCrawlResult** element the response message.

• On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.41.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.41.1.1 PauseCrawlSoapIn

This is the request message for **PauseCrawl** operation.

The SOAP action value of the message is defined as:

```
http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/PauseCrawl
```

The SOAP body contains a **PauseCrawl** element.

#### 3.1.4.41.1.2 PauseCrawlSoapOut

This is a response message for **PauseCrawl** operation.

The SOAP action value of the message is defined as:

```
http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/PauseCrawl
```

The SOAP body contains a **PauseCrawlResponse** element.

### 3.1.4.41.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.41.2.1 PauseCrawl

The definition of the **PauseCrawl** element is as follows:

```xml
<s:element name="PauseCrawl">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="contentSource" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

**contentSource:** The identifier of the content source for which to pause the crawl.

#### 3.1.4.41.2.2 PauseCrawlResponse

The definition of the **PauseCrawlResponse** element is as follows:

```xml
<s:element name="PauseCrawlResponse"/>
```
<s:complexType>
  <s:sequence>
    <s:element name="PauseCrawlResult" type="tns:ContentSourceDynamicPropsInternal"
      minOccurs="0"/>
  </s:sequence>
</s:complexType>

**PauseCrawlResult:** The status of the most recent crawl. This element MUST be present and MUST conform to the schema of the `ContentSourceDynamicPropsInternal` complex type.

### 3.1.4.42 RefreshAnchorContentSource

This method is used to retrieve the current status of the anchor content source of the portal content project.

```xml
<wsdl:operation name="RefreshAnchorContentSource">
  <wsdl:input message="RefreshAnchorContentSourceSoapIn"/>
  <wsdl:output message="RefreshAnchorContentSourceSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `RefreshAnchorContentSourceSoapIn` request message and the protocol server responds with a `RefreshAnchorContentSourceSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- On success, the protocol server returns the current anchor content source status in the `RefreshAnchorContentSourceResult` element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.42.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.42.1.1 RefreshAnchorContentSourceSoapIn

This is the request message for `RefreshAnchorContentSource` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `RefreshAnchorContentSource` element.

#### 3.1.4.42.1.2 RefreshAnchorContentSourceSoapOut

This is a response message for `RefreshAnchorContentSource` operation.

The SOAP action value of the message is defined as:
The SOAP body contains a **RefreshAnchorContentSourceResponse** element.

### 3.1.4.42.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.42.2.1 RefreshAnchorContentSource

The **RefreshAnchorContentSource** element represents parameters for the **RefreshAnchorContentSource** operation.

```xml
<s:element name="RefreshAnchorContentSource">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

#### 3.1.4.42.2.2 RefreshAnchorContentSourceResponse

The **RefreshAnchorContentSourceResponse** element represents the result of the **RefreshAnchorContentSource** operation.

```xml
<s:element name="RefreshAnchorContentSourceResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="RefreshAnchorContentSourceResult" type="tns:ContentSourceDynamicPropsInternal" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**RefreshAnchorContentSourceResult**: The status of the anchor content source. This element MUST be present and MUST conform to the schema of the **ContentSourceDynamicPropsInternal** complex type. The errorCount element in the **ContentSourceDynamicPropsInternal** element of the **RefreshAnchorContentSourceResult** element MUST contain 0. The element **crawlSuccesses** MUST contain 0.

### 3.1.4.43 RefreshContentSource

This operation is used to retrieve the current status of a content source from the portal content project.

```xml
<wsdl:operation name="RefreshContentSource">
  <wsdl:input message="RefreshContentSourceSoapIn"/>
  <wsdl:output message="RefreshContentSourceSoapOut"/>
</wsdl:operation>
```
The protocol client sends a **RefreshContentSourceSoapIn** request message and the protocol server responds with a **RefreshContentSourceSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrentException**.

- If the content source specified by the identifier in the request message does not exist in the portal content project, the protocol server MUST throw a SOAP fault message containing a **DeletedConcurrentException**.

- On success, the protocol server MUST return current content source status in the **RefreshContentSourceResult** element of the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.43.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.43.1.1 RefreshContentSourceSoapIn

This is the request message for **RefreshContentSource** operation.

The SOAP action value of the message is defined as:


The SOAP body contains a **RefreshContentSource** element.

#### 3.1.4.43.1.2 RefreshContentSourceSoapOut

This is a response message for **RefreshContentSource** operation.

The SOAP action value of the message is defined as:


The SOAP body contains a **RefreshContentSourceResponse** element.

### 3.1.4.43.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.43.2.1 RefreshContentSource

The **RefreshContentSource** element represents parameters for the **RefreshContentSource** operation.

```xml
<s:element name="RefreshContentSource">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="id" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
versionIn: The configuration version of the crawler application. This element MUST contain a value
greater than or equal to zero.

id: The content source identifier. MUST be greater than 0.

3.1.4.43.2.2 RefreshContentSourceResponse

The RefreshContentSourceResponse element represents the result of the RefreshContentSource
operation.

```xml
<s:element name="RefreshContentSourceResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="RefreshContentSourceResult" type="tns:ContentSourceDynamicPropsInternal" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

RefreshContentSourceResult: The status of the content source. This element MUST be present and
MUST conform to the schema of the ContentSourceDynamicPropsInternal complex type.

3.1.4.44 RemoveAllGathererApplications

This operation is not used and MUST NOT be invoked.

3.1.4.45 RemoveApp

This operation is used to remove a crawler application and the full-text index catalog belonging to it.

```xml
<wsdl:operation name="RemoveApp">
  <wsdl:input message="RemoveAppSoapIn"/>
  <wsdl:output message="RemoveAppSoapOut"/>
</wsdl:operation>
```

The protocol client sends a RemoveAppSoapIn request message and the protocol server responds
with a RemoveAppSoapOut response message, as follows:

- If a crawler application with the name specified by the applicationName element of the request
  message exists, the protocol server MUST remove the application, all the objects contained within
  the application, and the full-text index catalog belonging to the application.
- If the specified application is absent, the protocol server SHOULD NOT consider the absence a
  failure and SHOULD continue.
- If a folder exists at the location specified by the path resulting from combining the indexLocation
  element of the request message and the applicationName element of the request message, the
  protocol server MUST remove the folder as well.
- If the specified folder is absent, the server SHOULD NOT consider the absence a failure and
  SHOULD continue.
- On error, the protocol server MUST throw a SOAP fault message.
3.1.4.45.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.45.1.1 RemoveAppSoapIn

This is the request message for RemoveApp operation.
The SOAP action value of the message is defined as:


The SOAP body contains a RemoveApp element.

3.1.4.45.1.2 RemoveAppSoapOut

This is a response message for RemoveApp operation.
The SOAP action value of the message is defined as:


The SOAP body contains a RemoveAppResponse element.

3.1.4.45.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.45.2.1 RemoveApp

The definition of the RemoveApp element is as follows:

   <s:element name="RemoveApp">
       <s:complexType>
           <s:sequence>
               <s:element name="applicationName" type="s:string" minOccurs="0"/>
               <s:element name="indexLocation" type="s:string" minOccurs="0"/>
           </s:sequence>
       </s:complexType>
   </s:element>

applicationName: Name of the crawler application. This element MUST be present.

indexLocation: The path to the folder to be removed on the index server. This element MUST be present.

3.1.4.45.2.2 RemoveAppResponse

The definition of the RemoveAppResponse element is as follows:

   <s:element name="RemoveAppResponse">
       <s:complexType/>
   </s:element>
3.1.4.46  RemoveContentSource

This operation is used to remove a content source from the portal content project in the crawler application.

```xml
<wsdl:operation name="RemoveContentSource">
  <wsdl:input message="RemoveContentSourceSoapIn"/>
  <wsdl:output message="RemoveContentSourceSoapOut"/>
</wsdl:operation>
```

The protocol client sends a RemoveContentSourceSoapIn request message and the protocol server responds with a RemoveContentSourceSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- The protocol server MUST look up a content source by the identifier specified in the id element in the request message. If the content source does not exist, the protocol server MUST throw a SOAP fault message containing a DeletedConcurrencyException.
- The protocol server removes the content source.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the RemoveContentSourceResult element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.46.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.46.1.1  RemoveContentSourceSoapIn

This is the request message for removing RemoveContentSource operation.

The SOAP action value of the message is defined as:


The SOAP body contains a RemoveContentSource element.

3.1.4.46.1.2  RemoveContentSourceSoapOut

This is a response message for RemoveContentSource operation.

The SOAP action value of the message is defined as:


The SOAP body contains a RemoveContentSourceResponse element.
3.1.4.46.2 Elements
The following XML Schema element definitions are specific to this operation.

3.1.4.46.2.1 RemoveContentSource
The definition of the RemoveContentSource element is as follows:

```xml
<element name="RemoveContentSource">
  <complexType>
    <sequence>
      <element name="versionIn" type="s:int"/>
      <element name="catalog" type="s:int"/>
      <element name="id" type="s:int"/>
    </sequence>
  </complexType>
</element>
```

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

catalog: This element MUST contain the value 0.

id: This element MUST contain the content source identifier. The identifier MUST be greater than 0.

3.1.4.46.2.2 RemoveContentSourceResponse
The definition of the RemoveContentSourceResponse element is as follows:

```xml
<element name="RemoveContentSourceResponse">
  <complexType>
    <sequence>
      <element name="RemoveContentSourceResult" type="s:int"/>
    </sequence>
  </complexType>
</element>
```

RemoveContentSourceResult: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.47 RemoveCrawlMapping
This method is used to remove a crawl mapping from the portal content project.

```xml
<operation name="RemoveCrawlMapping">
  <input message="RemoveCrawlMappingSoapIn"/>
  <output message="RemoveCrawlMappingSoapOut"/>
</operation>
```

The protocol client sends a RemoveCrawlMappingSoapIn request message and the protocol server responds with a RemoveCrawlMappingSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.

- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- If no crawl mapping can be found which has an access URL matching the source element in the request message, the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040d06.

- If the crawl mapping exists, the protocol server MUST remove the crawl mapping from the portal content project.

- On success, the protocol server MUST return the updated configuration version of the crawler application in the RemoveCrawlMappingResult element in the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.47.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.47.1.1 RemoveCrawlMappingSoapIn

This is the request message for RemoveCrawlMapping operation.

The SOAP action value of the message is defined as:


The SOAP body contains a RemoveCrawlMapping element.

#### 3.1.4.47.1.2 RemoveCrawlMappingSoapOut

This is a response message for RemoveCrawlMapping operation.

The SOAP action value of the message is defined as:


The SOAP body contains a RemoveCrawlMappingResponse element.

### 3.1.4.47.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.47.2.1 RemoveCrawlMapping

The RemoveCrawlMapping element represents parameters for the RemoveCrawlMapping operation.

```xml
<s:element name="RemoveCrawlMapping">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="source" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

source: The access URL prefix. This element MUST be present and its content MUST be less than 2048 characters in length. It MUST match an access URL of an existing crawl mapping.

3.1.4.47.2.2 RemoveCrawlMappingResponse
The RemoveCrawlMappingResult element represents the result of the RemoveCrawlMapping operation.

```xml
<s:element name="RemoveCrawlMappingResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="RemoveCrawlMappingResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

RemoveCrawlMappingResult: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.48 RemoveCrawlRule
This operation removes a crawl rule from the portal content project.

```xml
<wsdl:operation name="RemoveCrawlRule">
  <wsdl:input message="RemoveCrawlRuleSoapIn"/>
  <wsdl:output message="RemoveCrawlRuleSoapOut"/>
</wsdl:operation>
```

The protocol client sends a RemoveCrawlRuleSoapIn request message and the protocol server responds with a RemoveCrawlRuleSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- The protocol server MUST normalize the path element of the request message the same way it normalizes the path element of the ActivateCrawlRule operation request message.
- Then, the protocol server MUST try to find an exact match for the crawl rule in the crawl rules collection of the portal content project by the normalized path.
- If the crawl rule is not found the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040d06.
- If the crawl rule is found, the protocol server MUST remove the crawl rule from the portal content project.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the RemoveCrawlRuleResult element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.
3.1.4.48.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.48.1.1 RemoveCrawlRuleSoapIn

This is the request message for the RemoveCrawlRule operation.

The SOAP action value of the message is defined as:


The SOAP body contains a RemoveCrawlRule element.

3.1.4.48.1.2 RemoveCrawlRuleSoapOut

This is a response message for the RemoveCrawlRule operation.

The SOAP action value of the message is defined as:


The SOAP body contains a RemoveCrawlRuleResponse element.

3.1.4.48.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.48.2.1 RemoveCrawlRule

The RemoveCrawlRule element contains the parameters for the RemoveCrawlRule operation.

```xml
<s:element name="RemoveCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="path" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

path: The path expression of the crawl rule which MUST be present and MUST have a length greater than 0 and less than 2048 characters.

3.1.4.48.2.2 RemoveCrawlRuleResponse

The RemoveCrawlRuleResponse element contains the result of the RemoveCrawlRule operation.

```xml
<s:element name="RemoveCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="RemoveCrawlRuleResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
**RemoveCrawlRuleResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

### 3.1.49 RemoveExtension

This operation is used to remove a file extension from the *extensions* list defined for the portal content project.

```xml
<wsdl:operation name="RemoveExtension">
    <wsdl:input message="RemoveExtensionSoapIn"/>
    <wsdl:output message="RemoveExtensionSoapOut"/>
</wsdl:operation>
```

The protocol client sends a *RemoveExtensionSoapIn* request message and the protocol server responds with a *RemoveExtensionSoapOut* response message, as follows:

- On receipt the protocol server validates the *versionIn* value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an *UpdatedConcurrencyException*.

- The protocol server then updates the current configuration version of the crawler application to the value of *versionIn*+1.

- If the file extension specified by ext element in the request message does not exist in the list, the protocol server MUST send a SOAP fault message containing a *COMException* with *HRESULT* field set to 0x80040d06.

- The protocol server removes a file extension specified by the ext element in the request message from extension list property of the portal content project.

- On success, the protocol server MUST return the updated configuration version of the crawler application in the *RemoveExtensionResult* element in the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.49.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.49.1.1 RemoveExtensionSoapIn

This is the request message for *RemoveExtension* operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains a *RemoveExtension* element.

#### 3.1.49.1.2 RemoveExtensionSoapOut
This is a response message for **RemoveExtension** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **RemoveExtensionResponse** element.

### 3.1.4.49.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.49.2.1 RemoveExtension

The definition of the **RemoveExtension** element is as follows:

```
<xs:element name="RemoveExtension">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="versionIn" type="xs:int"/>
      <xs:element name="ext" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST be present and MUST contain a value greater than or equal to zero.

**ext**: file extension. MUST be present. Length MUST be greater than 0 and less than 16 characters.

#### 3.1.4.49.2.2 RemoveExtensionResponse

The definition of the **RemoveExtensionResponse** element is as follows:

```
<xs:element name="RemoveExtensionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="RemoveExtensionResult" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**RemoveExtensionResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to 0.

### 3.1.4.50 RemoveQueryServer

This operation is not used and MUST NOT be invoked.

### 3.1.4.51 RemoveSitePath

This operation is not used and MUST NOT be invoked.

### 3.1.4.52 RemoveSiteRestriction

This operation is not used and MUST NOT be invoked.
3.1.4.53  ResetApp

This operation is used to reset the existing crawler application that is not dismounted on the index server. Any state acquired by the crawler application during previously conducted crawls, as well as all the full-text index catalog will get removed.

```
<wsdl:operation name="ResetApp">
  <wsdl:input message="ResetAppSoapIn"/>
  <wsdl:output message="ResetAppSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `ResetAppSoapIn` request message and the protocol server responds with a `ResetAppSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- The protocol server then updates the current configuration version of the crawler application to the value of `versionIn + 1`.
- Then, the protocol server removes all data acquired during all crawls within the crawler application.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the `ResetAppResult` element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.53.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.53.1.1  ResetAppSoapIn

This is the request message for `ResetApp` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ResetApp` element.

3.1.4.53.1.2  ResetAppSoapOut

This is a response message for `ResetApp` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ResetAppResponse` element.

3.1.4.53.2  Elements

The following XML Schema element definitions are specific to this operation.
3.1.4.53.2.1 ResetApp

The definition of the **ResetApp** element is as follows:

```xml
<s:element name="ResetApp">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.53.2.2 ResetAppResponse

The definition of the **ResetAppResponse** element is as follows:

```xml
<s:element name="ResetAppResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ResetAppResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**ResetAppResult:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.54 ResumeCrawl

This operation is used to resume a crawl of a content source of the portal content project.

```xml
<wsdl:operation name="ResumeCrawl">
  <wsdl:input message="ResumeCrawlSoapIn"/>
  <wsdl:output message="ResumeCrawlSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **ResumeCrawlSoapIn** request message and the protocol server responds with a **ResumeCrawlSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.
- If there is no content source with the identifier specified in the **contentSource** element of the request message, the protocol server MUST throw a SOAP fault message containing a **DeletedConcurrencyException**.
- The protocol server MUST try to resume a crawl for the content source with the identifier specified in the **contentSource** element of the request message.
- If the crawl is in a state where it cannot be resumed or is already resumed, the protocol server MUST ignore the resume request and return the current crawl status in the **ResumeCrawlResult** element the response message.
On success, the protocol server MUST return the current crawl status in the `ResumeCrawlResult` element the response message.

On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.54.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.54.1.1 ResumeCrawlSoapIn

This is the request message for `ResumeCrawl` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ResumeCrawl` element.

#### 3.1.4.54.1.2 ResumeCrawlSoapOut

This is a response message for `ResumeCrawl` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ResumeCrawlResponse` element.

### 3.1.4.54.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.54.2.1 ResumeCrawl

The definition of the `ResumeCrawl` element is as follows:

```xml
<s:element name="ResumeCrawl">
    <s:complexType>
        <s:sequence>
            <s:element name="versionIn" type="s:int"/>
            <s:element name="contentSource" type="s:int"/>
        </s:sequence>
    </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**contentSource**: The identifier of the content source for which to resume the crawl.

#### 3.1.4.54.2.2 ResumeCrawlResponse

The definition of the `ResumeCrawlResponse` element is as follows:

```xml
<s:element name="ResumeCrawlResponse">
```

---

[MS-ADMWS] - v20171212
Search Service Administration Web Service Protocol
Copyright © 2017 Microsoft Corporation
Release: December 12, 2017
<s:complexType>
<s:sequence>
<s:element name="ResumeCrawlResult" type="tns:ContentSourceDynamicPropsInternal" minOccurs="0"/>
</s:sequence>
</s:complexType>

ResumeCrawlResult: The status of the most recent crawl. This element MUST be present and MUST conform to the schema of ContentSourceDynamicPropsInternal complex type.

3.1.4.55 SearchDatabaseCleanup

This operation is used to enable or disable the search database clean up step during service startup. Clean up step SHOULD include operations for optimization of the search database such as deleting unused data, and so on. The complete list of these operations is up to the protocol server implementation.

```xml
<wsdl:operation name="SearchDatabaseCleanup">
  <wsdl:input message="SearchDatabaseCleanupSoapIn"/>
  <wsdl:output message="SearchDatabaseCleanupSoapOut"/>
</wsdl:operation>
```

The protocol client sends a SearchDatabaseCleanupSoapIn request message and the protocol server responds with a SearchDatabaseCleanupSoapOut response message, as follows:

- The protocol server MUST enable or disable search database clean up according to the value of the enable element of the request message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.55.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.55.1.1 SearchDatabaseCleanupSoapIn

This is the request message for SearchDatabaseCleanup operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a SearchDatabaseCleanup element.

3.1.4.55.1.2 SearchDatabaseCleanupSoapOut

This is a response message for SearchDatabaseCleanup operation.

The SOAP action value of the message is defined as:

```
```
The SOAP body contains a **SearchDatabaseCleanupResponse** element.

### 3.1.4.55.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.55.2.1 SearchDatabaseCleanup

The definition of the **SearchDatabaseCleanup** element is as follows:

```
<xs:element name="SearchDatabaseCleanup">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="versionIn" type="xs:int"/>
      <xs:element name="enable" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**enable**: If true, indicates the search database clean up step during service startup is enabled, otherwise disabled.

#### 3.1.4.55.2.2 SearchDatabaseCleanupResponse

The definition of the **SearchDatabaseCleanupResponse** element is as follows:

```
<xs:element name="SearchDatabaseCleanupResponse">
  <xs:complexType/>
</xs:element>
```

### 3.1.4.56 SetContentSourcesMetadata

This operation is used to set the metadata property associated with the portal content project. This metadata string intended for protocol client use only, the protocol server just stores it without interpreting. Once set, metadata string can be obtained by calling **GetContentSources** operation.

```
<wsdl:operation name="SetContentSourcesMetadata">
  <wsdl:input message="SetContentSourcesMetadataSoapIn"/>
  <wsdl:output message="SetContentSourcesMetadataSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **SetContentSourcesMetadataSoapIn** request message and the protocol server responds with a **SetContentSourcesMetadataSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.
- The protocol server then updates the current configuration version of the crawler application to the value of **versionIn+1**.
- The protocol server sets the **contentSourcesMetadata** property of the **portal content project** with the metadata value specified in the request message.
On success, the protocol server MUST return the updated configuration version of the crawler application in the `SetContentSourcesMetadataResult` element in the response message.

On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.56.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.56.1.1 SetContentSourcesMetadataSoapIn

This is the request message for `SetContentSourcesMetadata` operation.

The SOAP action value of the message is defined as:

```
http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/SetContentSourcesMetadata
```

The SOAP body contains a `SetContentSourcesMetadata` element.

#### 3.1.4.56.1.2 SetContentSourcesMetadataSoapOut

This is a response message for `SetContentSourcesMetadata` operation.

The SOAP action value of the message is defined as:

```
http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService/SetContentSourcesMetadata
```

The SOAP body contains a `SetContentSourcesMetadataResponse` element.

### 3.1.4.56.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.56.2.1 SetContentSourcesMetadata

The `SetContentSourcesMetadata` element represents parameters for the `SetContentSourcesMetadata` operation.

```xml
<xs:element name="SetContentSourcesMetadata">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="versionIn" type="xs:int"/>
      <xs:element name="metadata" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

- **versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

- **metadata**: Arbitrary metadata associated by the protocol client with the content source. If present, the length MUST be less than 1024 characters. The value of this property is ignored by the protocol server, but can be interpreted by the protocol client to associate arbitrary metadata with the content source collection.
3.1.4.56.2.2 SetContentSourcesMetadataResponse

The **SetContentSourcesMetadataResponse** element represents the result for the **SetContentSourcesMetadata** operation.

```xml
<s:element name="SetContentSourcesMetadataResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetContentSourcesMetadataResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**SetContentSourcesMetadataResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.57 SetCrawlRuleCredentials

This operation is used to configure the authentication method and crawl account for a crawl rule.

```xml
<wsdl:operation name="SetCrawlRuleCredentials">
  <wsdl:input message="SetCrawlRuleCredentialsSoapIn"/>
  <wsdl:output message="SetCrawlRuleCredentialsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **SetCrawlRuleCredentialsSoapIn** request message and the protocol server responds with a **SetCrawlRuleCredentialsSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.
- The protocol server then updates the current configuration version of the crawler application to the value of **versionIn+1**.
- The protocol server MUST normalize the path element of the request message the same way it normalizes the path element of the **ActivateCrawlRule** operation request message.
- If there is no crawl rule in the crawl rule collection which has a URL equal to the normalized path, the protocol server MUST throw SOAP fault message containing **DeletedConcurrencyException**.
- The protocol server updates the **authType** property of the crawl rule from the corresponding parameter of the method.
  - If the **authType** is 0, the protocol server MUST set the **accountName** property of the crawl rule to empty string, and the **password** property of the crawl rule to an empty string.
  - If the **authType** is 1, the protocol server MUST set the **accountName** property of the crawl rule to the value of **authString1**, and the **password** property of the crawl rule to the value of **authString2**. The protocol server MUST validate the correctness of the credentials on the index server. If credentials are invalid, the protocol server MUST throw SOAP fault message containing a **COMException**.
  - If the authType is 2, the protocol server MUST set the **accountName** property of the crawl rule to the value of **authString1** element, and the **password** property of the crawl rule to the value of **authString2** element.
  - If the authType is 3, the protocol server MUST set the **certificateName** property of the crawl rule to the value of **authString1** element.
- If the authType is 4, the protocol server MUST set the password property of the crawl rule to the value of the authString1 element. Then, the protocol server MUST extract and sets the authUrl, authData, authMethod and miscData properties of the crawl rule from the value of the authString2 element.

- If the authType is 5, the protocol server MUST set the cookieName property of the crawl rule to the value of the authString1 element, and sets the miscData property of the crawl rule to the value of the authString2 element.

- On success, the protocol server MUST return the updated configuration version of the crawler application in the SetCrawlRuleCredentialsResult element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.57.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.57.1.1 SetCrawlRuleCredentialsSoapIn

This is the request message for SetCrawlRuleCredentials operation.

The SOAP action value of the message is defined as:

Credentials

The SOAP body contains a SetCrawlRuleCredentials element.

3.1.4.57.1.2 SetCrawlRuleCredentialsSoapOut

This is a response message for SetCrawlRuleCredentials operation.

The SOAP action value of the message is defined as:

Credentials

The SOAP body contains a SetCrawlRuleCredentialsResponse element.

3.1.4.57.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.57.2.1 SetCrawlRuleCredentials

The SetCrawlRuleCredentials element represents the parameters for the SetCrawlRuleCredentials operation.

```xml
<element name="SetCrawlRuleCredentials">
  <complexType>
    <sequence>
      <element name="versionIn" type="s:int"/>
      <element name="path" type="s:string" minOccurs="0"/>
      <element name="authType" type="s:int"/>
      <element name="authString1" type="s:string" minOccurs="0"/>
      <element name="authString2" type="s:string" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
```
versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

path: The path expression corresponding to the URL of the crawl rule. MUST be present and length MUST be greater than 0 and less than 2048 characters. It MUST correspond to a path of a crawl rule in the crawler application’s crawl rule collection.

authType: The type of the authentication to access matching URLs. MUST have one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default access</td>
</tr>
<tr>
<td>1</td>
<td>Integrated Windows authentication</td>
</tr>
<tr>
<td>2</td>
<td>Basic authentication</td>
</tr>
<tr>
<td>3</td>
<td>Authentication using certificates</td>
</tr>
<tr>
<td>4</td>
<td>Forms authentication</td>
</tr>
<tr>
<td>5</td>
<td>Cookie based authentication</td>
</tr>
</tbody>
</table>

Default access implies integrated authentication using credentials of the default crawl account for the crawler application.

authString1:

- If authType is 0, the protocol server MUST ignore this element.
- If authType is 1 or 2, the value of the element represents the user name, and the length MUST be less than 256 characters.
- If authType is 3, this element represents a certificate name.
- If authType is 4, this element represents the portion of the forms authentication request data that MUST be stored securely by the protocol server.
- If authType is 5, this element represents the cookie for the cookie based authentication.

authString2:

- If authType is 0 or 3, the protocol server MUST ignore this element.
- If authType is 1 or 2, this element represents the password for the account specified in authString1.
- If authType is 4, this element contains the information corresponding to authUrl, authData, miscData, and authMethod properties of the of the crawl rule used for forms authentication in the format specified in the following table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>authString2</td>
<td>auth_url_and_method &quot;;&quot; requestParams &quot;;&quot;error_pages</td>
</tr>
<tr>
<td>Property</td>
<td>Format</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>auth_url_and_method</td>
<td>method_flag requestUrl</td>
</tr>
<tr>
<td>method_flag</td>
<td>&quot;&quot; / &quot;p$&quot; / &quot;g$&quot; / &quot;P$&quot; / &quot;G$&quot;</td>
</tr>
<tr>
<td>error_pages</td>
<td>*( errorPage &quot;;;&quot; )</td>
</tr>
</tbody>
</table>

- **requestUrl** corresponds to the **authUrl** property of the of the crawl rule.
- **requestParams** corresponds to the **authData** property of the of the crawl rule.
- **error_pages** corresponds to the **miscData** property of the of the crawl rule.
- Empty, "p$", and "P$" method_flag values correspond to a HTTP POST authMethod property of the of the crawl rule.
- "g$", and "G$" method_flag values correspond to a HTTP GET authMethod property of the of the crawl rule.
- To perform forms authentication, the protocol server sends request data (format: requestParams ";& authString1) to the resource located at requestUrl using either HTTP GET or HTTP POST depending on the value of method_flag.

The following example illustrates the format of this element:

```
g$http://www.contoso.com:56737/login.aspx;<requestParams>;<errorPage1>;<errorPage2>
```

If authType is 5, this element represents a list of error pages URLs in the format specified in the following table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>authString2</td>
<td>&quot;;;&quot; error_pages</td>
</tr>
<tr>
<td>error_pages</td>
<td>*( errorPage &quot;;;&quot; )</td>
</tr>
</tbody>
</table>

The following example illustrates the format of this element:

```
;;<errorPage1>;<errorPage2>
```

### 3.1.4.57.2.2 SetCrawlRuleCredentialsResponse

The **SetCrawlRuleCredentialsResponse** element represents the result of the **SetCrawlRuleCredentials** operation.

```
<s:element name="SetCrawlRuleCredentialsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetCrawlRuleCredentialsResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**SetCrawlRuleCredentialsResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.
3.1.4.58  SetCrawlRulePriority

This operation is used to modify the order of the crawl rules in the ordered collection of the crawl rules in the portal content project.

```xml
<wsdl:operation name="SetCrawlRulePriority">
  <wsdl:input message="SetCrawlRulePrioritySoapIn"/>
  <wsdl:output message="SetCrawlRulePrioritySoapOut"/>
</wsdl:operation>
```

The protocol client sends a SetCrawlRulePrioritySoapIn request message and the protocol server responds with a SetCrawlRulePrioritySoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
- The protocol server then updates the current configuration version of the crawler application to the value of versionIn+1.
- If the currentPriority is the same as the newPriority, the protocol server MUST return the updated configuration version of the crawler application in the SetCrawlRulePriorityResult element in the response message.
- If the priorities are different, the protocol server MUST try to find a match for the path element’s value against the crawl rules in the crawl rules collection of the portal content project.
- If no matching crawl rule is found the protocol server MUST throw a SOAP fault message containing a COMException with the HRESULT field set to 0x80040d06.
- If the crawl rule is found, the protocol server moves the crawl rule to the position computed by adding the value obtained by subtracting currentPriority from newPriority to the current index of the crawl rule.
  - If the new index is greater than the size of the collection, the crawl rule MUST be inserted last.
  - If the new index is less than 0 the crawl rule MUST be inserted at the first position.
  - Otherwise, the crawl rule MUST be inserted at the new index position in the crawl rules collection.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the SetCrawlRulePriorityResult element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.58.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.58.1.1  SetCrawlRulePrioritySoapIn

This is the request message for the SetCrawlRulePriority operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a SetCrawlRulePriority element.
### 3.1.4.58.1.2 SetCrawlRulePrioritySoapOut

This is a response message for the `SetCrawlRulePriority` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `SetCrawlRulePriorityResponse` element.

### 3.1.4.58.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.58.2.1 SetCrawlRulePriority

The `SetCrawlRulePriority` element contains parameters of the `SetCrawlRulePriority` operation.

```xml
<s:element name="SetCrawlRulePriority">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="path" type="s:string" minOccurs="0"/>
      <s:element name="currentPriority" type="s:int"/>
      <s:element name="newPriority" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**path:** The path expression corresponding to the URL of the crawl rule. MUST be present and MUST have length between 0 and 2048.

**currentPriority:** Any value. The difference between the `newPriority` element and the `currentPriority` element is used as an offset to the current position of the crawl rule to modify its priority.

**newPriority:** Any value. The difference between the `newPriority` element and the `currentPriority` element is used as an offset to the current position of the crawl rule to modify its priority.

#### 3.1.4.58.2.2 SetCrawlRulePriorityResponse

The `SetCrawlRulePriorityResponse` element contains the result of the `SetCrawlRulePriority` operation.

```xml
<s:element name="SetCrawlRulePriorityResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetCrawlRulePriorityResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**SetCrawlRulePriorityResult:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.
3.1.4.59  SetDefaultGatheringAccount

This operation is used to set the default crawl account for the crawler application.

```xml
<wsdl:operation name="SetDefaultGatheringAccount">
  <wsdl:input message="SetDefaultGatheringAccountSoapIn"/>
  <wsdl:output message="SetDefaultGatheringAccountSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `SetDefaultGatheringAccountSoapIn` request message and the protocol server responds with a `SetDefaultGatheringAccountSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.

- The protocol server then updates the current configuration version of the crawler application to the value of `versionIn+1`.

- If the value of the account element in the request message is NULL or empty, the protocol server MUST set the `defaultCrawlingAccount` properties of the crawler application to empty. The protocol server then MUST use the search service account as the default crawl account.

- If the value of the account element in the request message is not a valid user name or if an user with such a name does not exist, the protocol server MUST throw a SOAP fault message containing a `COMException`.

- If the value of the password element in the request message is not the correct password for the non-NULL and non-empty value of the account element in the request message, the protocol server MUST throw SOAP fault message containing a `COMException`.

- Otherwise, the protocol server sets the `defaultCrawlingAccount` and `defaultCrawlingAccountPassword` properties of the crawler application to the account and password values from the request message respectively.

- On success, the protocol server MUST return the updated configuration version of the crawler application in the `SetDefaultGatheringAccountResult` element in the response message.

- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.59.1  Messages

The following WSDL message definitions are specific to this operation.

3.1.4.59.1.1  SetDefaultGatheringAccountSoapIn

This is the request message for `SetDefaultGatheringAccount` operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains a `SetDefaultGatheringAccount` element.

3.1.4.59.1.2  SetDefaultGatheringAccountSoapOut

This is a response message for `SetDefaultGatheringAccount` operation.
The SOAP action value of the message is defined as:


The SOAP body contains a SetDefaultGatheringAccountResponse element.

3.1.4.59.2  Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.59.2.1  SetDefaultGatheringAccount

The definition of the SetDefaultGatheringAccount element is as follows:

```xml
<xs:element name="SetDefaultGatheringAccount">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="versionIn" type="xs:int"/>
      <xs:element name="account" type="xs:string" minOccurs="0"/>
      <xs:element name="password" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

account: The user name. This element MUST be present, and the length MUST be less than 256 characters.

password: The password for the account. This element MUST be present.

3.1.4.59.2.2  SetDefaultGatheringAccountResponse

The definition of the SetDefaultGatheringAccountResponse element is as follows:

```xml
<xs:element name="SetDefaultGatheringAccountResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SetDefaultGatheringAccountResult" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

SetDefaultGatheringAccountResult: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

3.1.4.60  SetDefaultImportAccount

This operation is used to set the default user profile import account for the user profile import project.

```xml
<wsdl:operation name="SetDefaultImportAccount">
  <wsdl:input message="SetDefaultImportAccountSoapIn"/>
  <wsdl:output message="SetDefaultImportAccountSoapOut"/>
</wsdl:operation>
```
The protocol client sends a `SetDefaultImportAccountSoapIn` request message and the protocol server responds with a `SetDefaultImportAccountSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.

- The protocol server then updates the current configuration version of the crawler application to the value of `versionIn+1`.

- If the value of the `userID` element in the request message is NULL or empty, the protocol server MUST set both the `defaultAccountName` and `defaultAccountPassword` properties of the user profile import project to empty and will use **anonymous authentication** during the user profile import.

- Otherwise, the protocol server sets the `defaultAccountName` and `defaultAccountPassword` properties of the user profile import project to the `userID` and password values from the request message respectively.

- On success, the protocol server MUST return the updated configuration version of the crawler application in the `SetDefaultImportAccountResult` element in the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.60.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.60.1.1 SetDefaultImportAccountSoapIn

This is the request message for `SetDefaultImportAccount` operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains a `setDefaultImportAccount` element.

#### 3.1.4.60.1.2 SetDefaultImportAccountSoapOut

This is a response message for `SetDefaultImportAccount` operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains a `SetDefaultImportAccountResponse` element.

### 3.1.4.60.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.60.2.1 SetDefaultImportAccount

The definition of the `SetDefaultImportAccount` element is as follows:
<s:element name="SetDefaultImportAccount">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="userID" type="s:string" minOccurs="0"/>
      <s:element name="password" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**userID**: The user name used for authenticating the index server during the user profile import. This element MUST be present.

**password**: The default user profile import account password. This element MUST be present.

### 3.1.4.60.2.2 SetDefaultImportAccountResponse

The definition of the **SetDefaultImportAccountResponse** element is as follows:

<s:element name="SetDefaultImportAccountResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetDefaultImportAccountResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

**SetDefaultImportAccountResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

### 3.1.4.61 SetImportAccount

This operation is used to set the user profile import account of a user profile import domain for the user profile import project.

<wSDL:operation name="SetImportAccount">
  <wSDL:input message="SetImportAccountSoapIn"/>  
  <wSDL:output message="SetImportAccountSoapOut"/>
</wSDL:operation>

The protocol client sends a **SetImportAccountSoapIn** request message and the protocol server responds with a **SetImportAccountSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.
- The protocol server then updates the current configuration version of the crawler application to the value of **versionIn**+1.
- If the import domain specified by the **domain** element of the request message does not exist in the user profile import project, the protocol server MUST create it first.
- If the value of the **userID** element in the request message is NULL or empty, the protocol server sets both the **accountName** and **accountPassword** properties of the **user profile import domain**.
specified by the domain element of the request message to NULL and will use anonymous authentication during the user profile import of that user profile import domain.

- Otherwise, the protocol server sets the accountName and accountPassword properties of the user profile import domain specified by the domain element of the request message to the userID and password element values of the request message respectively.

- On success, the protocol server MUST return the updated configuration version of the crawler application in the SetImportAccountResult element in the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.61.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.61.1.1 SetImportAccountSoapIn

This is the request message for SetImportAccount operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a SetImportAccount element.

#### 3.1.4.61.1.2 SetImportAccountSoapOut

This is a response message for SetImportAccount operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a SetImportAccountResponse element.

### 3.1.4.61.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.61.2.1 SetImportAccount

The definition of the SetImportAccount element is as follows:

```xml
<s:element name="SetImportAccount">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="domain" type="s:string" minOccurs="0"/>
      <s:element name="userID" type="s:string" minOccurs="0"/>
      <s:element name="password" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```
**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**domain**: The domain name for which to set the user profile import account. This element MUST be present.

**userID**: The user name used for authentication of the index server during the user profile import. This element MUST be present.

**password**: The user profile import account password. This element MUST be present.

### 3.1.4.61.2.2 SetImportAccountResponse

The definition of the `SetImportAccountResponse` element is as follows:

```xml
<s:element name="SetImportAccountResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetImportAccountResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**SetImportAccountResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

### 3.1.4.62 SetIsExtensionIncludeList

This operation is used to set whether the list of file extensions in the portal content project is an inclusion list or an exclusion list.

```xml
<wsdl:operation name="SetIsExtensionIncludeList">
  <wsdl:input message="SetIsExtensionIncludeListSoapIn"/>
  <wsdl:output message="SetIsExtensionIncludeListSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `SetIsExtensionIncludeListSoapIn` request message and the protocol server responds with a `SetIsExtensionIncludeListSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- The protocol server then updates the current configuration version of the crawler application to the value of `versionIn+1`.
- The protocol server sets the `isExtensionIncludeList` property of the portal content project to the `isInclude` value from a request message.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the `SetIsExtensionIncludeListResponse` element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.62.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.62.1.1 SetIsExtensionIncludeListSoapIn
This is the request message for SetIsExtensionIncludeList operation.

The SOAP action value of the message is defined as:


The SOAP body contains a SetIsExtensionIncludeList element.

3.1.4.62.1.2 SetIsExtensionIncludeListSoapOut

This is a response message for SetIsExtensionIncludeList operation.

The SOAP action value of the message is defined as:


The SOAP body contains a SetIsExtensionIncludeListResponse element.

3.1.4.62.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.62.2.1 SetIsExtensionIncludeList

The definition of the SetIsExtensionIncludeList element is as follows:

<s:element name="SetIsExtensionIncludeList">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="isInclude" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

isInclude: The type of the file extensions list. If true, the file extensions list is an inclusion list; otherwise, the file extensions list is an exclusion list.

3.1.4.62.2.2 SetIsExtensionIncludeListResponse

The definition of the SetIsExtensionIncludeListResponse element is as follows:

<s:element name="SetIsExtensionIncludeListResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetIsExtensionIncludeListResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
**SetIsExtensionIncludeListResult**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

### 3.1.4.63 SetRetryLimit

This operation is used to set the retry limit for the crawler application.

```xml
<wsdl:operation name="SetRetryLimit">
  <wsdl:input message="SetRetryLimitSoapIn"/>
  <wsdl:output message="SetRetryLimitSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `SetRetryLimitSoapIn` request message and the protocol server responds with a `SetRetryLimitSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- The protocol server then updates the current configuration version of the crawler application to the value of `versionIn+1`.
- The protocol server sets the `retryLimit` property of the crawler application to the `retryLimit` value from the request message.
- On success, the protocol server MUST return the updated configuration version of the crawler application in the `SetRetryLimitResult` element in the response message.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.63.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.63.1.1 SetRetryLimitSoapIn

This is the request message for `SetRetryLimit` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `SetRetryLimit` element.

#### 3.1.4.63.1.2 SetRetryLimitSoapOut

This is a response message for `SetRetryLimit` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `SetRetryLimitResponse` element.

### 3.1.4.63.2 Elements
The following XML Schema element definitions are specific to this operation.

### 3.1.4.63.2.1 SetRetryLimit

The definition of the `SetRetryLimit` element is as follows:

```xml
<s:element name="SetRetryLimit">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="retryLimit" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**retryLimit:** The number of times to retry, MUST be greater than or equal to zero.

### 3.1.4.63.2.2 SetRetryLimitResponse

The definition of the `SetRetryLimitResponse` element is as follows:

```xml
<s:element name="SetRetryLimitResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetRetryLimitResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**SetRetryLimitResult:** The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

### 3.1.4.64 StartCrawl

This operation is used to begin a crawl of a content source of the portal content project.

```xml
<wsd1:operation name="StartCrawl">
  <wsdl:input message="StartCrawlSoapIn"/>
  <wsdl:output message="StartCrawlSoapOut"/>
</wsdl:operation>
```

The protocol client sends a `StartCrawlSoapIn` request message and the protocol server responds with a `StartCrawlSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- If there is no content source with the identifier specified in the `contentSource` element of the request message, the protocol server MUST throw a SOAP fault message containing a `DeletedConcurrencyException`.
- If a full crawl is specified in the request message’s `type` element, the protocol server MUST attempt to begin a full crawl for the specified content source.
• If an incremental crawl is specified in the request message’s type element, and the specified content source has not been crawled before, or the last crawl of the content source was stopped before it finished, the protocol server MUST attempt to start a full crawl for the content source.

• If an incremental crawl is specified in the request message’s type element, and the specified content source has been crawled before, and the last crawl was not stopped before it finished, the protocol server MUST attempt to start an incremental crawl for the content source.

• If the crawl is in a state where it cannot be started or is already started, the protocol server MUST ignore the start request and return the current crawl status in the StartCrawlResult element the response message.

• On success, the protocol server MUST return the current crawl status in the StartCrawlResult element of the response message.

• On error, the protocol server MUST throw a SOAP fault message.

3.1.4.64.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.64.1.1 StartCrawlSoapIn

This is the request message for StartCrawl operation.

The SOAP action value of the message is defined as:


The SOAP body contains a StartCrawl element.

3.1.4.64.1.2 StartCrawlSoapOut

This is a response message for StartCrawl operation.

The SOAP action value of the message is defined as:


The SOAP body contains a StartCrawlResponse element.

3.1.4.64.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.64.2.1 StartCrawl

The definition of the StartCrawl element is as follows:

```xml
<s:element name="StartCrawl">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="contentSource" type="s:int"/>
      <s:element name="type" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

contentSource: The identifier of the content source for which to start the crawl.

type: The type of the crawl to start. If 0, a full crawl; otherwise, an incremental crawl.

3.1.4.64.2.2 StartCrawlResponse

The definition of the StartCrawlResponse element is as follows:

```
<xs:element name="StartCrawlResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="StartCrawlResult" type="tns:ContentSourceDynamicPropsInternal" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

StartCrawlResult: The status of the most recent crawl. This element MUST be present and MUST conform to the schema of the ContentSourceDynamicPropsInternal complex type.

3.1.4.65 StartFullImport

This operation is used to begin a full user profile import of the specified user profile content source.

```
<wSDL:operation name="StartFullImport">
  <wSDL:input message="StartFullImportSoapIn"/>
  <wSDL:output message="StartFullImportSoapOut"/>
</wSDL:operation>
```

The protocol client sends a StartFullImportSoapIn request message and the protocol server responds with a StartFullImportSoapOut response message, as follows:

- The protocol server begins the full user profile import for the user profile content source specified by the contentSourceName element of the request message.
- If the user profile import is in a state where it cannot be started or is already started, the protocol server MUST ignore the start request.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.65.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.65.1.1 StartFullImportSoapIn

This is the request message for StartFullImport operation.

The SOAP action value of the message is defined as:
The SOAP body contains a **StartFullImport** element.

### 3.1.4.65.1.2 StartFullImportSoapOut

This is a response message for **StartFullImport** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **StartFullImportResponse** element.

### 3.1.4.65.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.65.2.1 StartFullImport

The definition of the **StartFullImport** element is as follows:

```xml
<s:element name="StartFullImport">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="contentSourceName" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn:** This value MUST be ignored by the protocol server.

**contentSourceName:** The user profile content source for which to start the full user profile import. This element MUST be present. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEOPLE_IMPORT</td>
<td>Start the user profile import.</td>
</tr>
<tr>
<td>PEOPLE_DL_IMPORT</td>
<td>Start the distribution list import.</td>
</tr>
</tbody>
</table>

#### 3.1.4.65.2.2 StartFullImportResponse

The definition of the **StartFullImportResponse** element is as follows:

```xml
<s:element name="StartFullImportResponse">
  <s:complexType/>
</s:element>
```
3.1.4.66 StartIncrementalImport

This operation is used to begin an incremental user profile import of the specified user profile content source.

<wsdl:operation name="StartIncrementalImport">
  <wsdl:input message="StartIncrementalImportSoapIn"/>
  <wsdl:output message="StartIncrementalImportSoapOut"/>
</wsdl:operation>

The protocol client sends a StartIncrementalImportSoapIn request message and the protocol server responds with a StartIncrementalImportSoapOut response message, as follows:

- The protocol server begins the incremental user profile import of the user profile content source specified by the contentSourceName element of the request message.
- If the user profile import is in a state where it cannot be started or is already started, the protocol server MUST ignore the start request.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.66.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.66.1.1 StartIncrementalImportSoapIn

This is the request message for StartIncrementalImport operation.

The SOAP action value of the message is defined as:


The SOAP body contains a StartIncrementalImport element.

3.1.4.66.1.2 StartIncrementalImportSoapOut

This is a response message for StartIncrementalImport operation.

The SOAP action value of the message is defined as:


The SOAP body contains a StartIncrementalImportResponse element.

3.1.4.66.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.66.2.1 StartIncrementalImport

The definition of the StartIncrementalImport element is as follows:

<s:element name="StartIncrementalImport">
<s:complexType>
<s:sequence>
<s:element name="versionIn" type="s:int"/>
<s:element name="contentSourceName" type="s:string" minOccurs="0"/>
</s:sequence>
</s:complexType>
</s:element>

**versionIn**: This value MUST be ignored by the protocol server.

**contentSourceName**: The user profile content source name for which to start the incremental user profile import. This element MUST be present. MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEOPLE_IMPORT</td>
<td>Start the user profile import.</td>
</tr>
<tr>
<td>PEOPLE_DL_IMPORT</td>
<td>Start the distribution list import.</td>
</tr>
</tbody>
</table>

### 3.1.4.66.2.2 StartIncrementalImportResponse

The definition of the **StartIncrementalImportResponse** element is as follows:

```xml
<s:element name="StartIncrementalImportResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.67 StartRankingUpdate

This operation is used to start the anchor crawl of the anchor content source.

```xml
<wSDL:operation name="StartRankingUpdate">
  <wSDL:input message="StartRankingUpdateSoapIn"/>
  <wSDL:output message="StartRankingUpdateSoapOut"/>
</wSDL:operation>
```

The protocol client sends a **StartRankingUpdateSoapIn** request message and the protocol server responds with a **StartRankingUpdateSoapOut** response message, as follows:

- The protocol server starts anchor crawl of the anchor content source.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.67.1 Messages

The following WSDL message definitions are specific to this operation.

### 3.1.4.67.1.1 StartRankingUpdateSoapIn

This is the request message for **StartRankingUpdate** operation.

The SOAP action value of the message is defined as:
The SOAP body contains a **StartRankingUpdate** element.

### 3.1.4.67.1.2 StartRankingUpdateSoapOut

This is a response message for **StartRankingUpdate** operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains a **StartRankingUpdateResponse** element.

### 3.1.4.67.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.67.2.1 StartRankingUpdate

The definition of the **StartRankingUpdate** element is as follows:

```xml
<xs:element name="StartRankingUpdate">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="versionIn" type="xs:int"/>
      <xs:element name="type" type="tns:RankingUpdateType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**versionIn**: The configuration version of the crawler application. This element **MUST** contain a value greater than or equal to zero.

**type**: This element **MUST** conform to the schema of **RankingUpdateType** simple type.

#### 3.1.4.67.2.2 StartRankingUpdateResponse

The definition of the **StartRankingUpdateResponse** element is as follows:

```xml
<xs:element name="StartRankingUpdateResponse">
  <xs:complexType/>
</xs:element>
```

#### 3.1.4.67.3 Complex Types

None.

#### 3.1.4.67.4 Simple Types

The following XML Schema simple definitions are specific to this operation.

#### 3.1.4.67.4.1 RankingUpdateType
The definition of the **RankingUpdateType** type is as follows:

```xml
<s:simpleType name="RankingUpdateType">
<s:restriction base="s:string">
<s:enumeration value="FullUpdate"/>
</s:restriction>
</s:simpleType>
```

**FullUpdate** is the only allowable values for this type.

### 3.1.4.68 StopCrawl

This operation is used to stop a crawl of a content source of the portal content project.

```xml
<wsdl:operation name="StopCrawl">
  <wsdl:input message="StopCrawlSoapIn"/>
  <wsdl:output message="StopCrawlSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **StopCrawlSoapIn** request message and the protocol server responds with a **StopCrawlSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.
- If there is no content source with the identifier specified in the **contentSource** element of the request message, the protocol server MUST throw SOAP fault message containing a **DeletedConcurrencyException**.
- The protocol server MUST try to stop the crawl of the content source with the identifier specified in the **contentSource** element of the request message.
- If the crawl is in a state where it cannot be stopped or is already stopped, the protocol server MUST ignore the stop request and return the current crawl status in the **StopCrawlResult** element the response message.
- On success, the protocol server MUST return the current crawl status in the **StopCrawlResult** element the response message.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.68.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.68.1.1 StopCrawlSoapIn

This is the request message for **StopCrawl** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **StopCrawl** element.
3.1.4.68.1.2 StopCrawlSoapOut

This is a response message for StopCrawl operation.

The SOAP action value of the message is defined as:


The SOAP body contains a StopCrawlResponse element.

3.1.4.68.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.68.2.1 StopCrawl

The definition of the StopCrawl element is as follows:

   <s:element name="StopCrawl">
     <s:complexType>
       <s:sequence>
         <s:element name="versionIn" type="s:int"/>
         <s:element name="contentSource" type="s:int"/>
       </s:sequence>
     </s:complexType>
   </s:element>

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

contentSource: The identifier of the content source for which to stop the crawl.

3.1.4.68.2.2 StopCrawlResponse

The definition of the StopCrawlResponse element is as follows:

   <s:element name="StopCrawlResponse">
     <s:complexType>
       <s:sequence>
         <s:element name="StopCrawlResult" type="tns:ContentSourceDynamicPropsInternal" minOccurs="0"/>
       </s:sequence>
     </s:complexType>
   </s:element>

StopCrawlResult: The status of the most recent crawl. This element MUST be present and MUST conform to the schema of the ContentSourceDynamicPropsInternal complex type.

3.1.4.69 StopImport

This operation is used to stop a user profile import of the specified user profile content source.

   <wsdl:operation name="StopImport">
     <wsdl:input message="StopImportSoapIn"/>
     <wsdl:output message="StopImportSoapOut"/>
   </wsdl:operation>
The protocol client sends a **StopImportSoapIn** request message and the protocol server responds with a **StopImportSoapOut** response message, as follows:

- The protocol server stops the user profile import of the user profile content source specified by the **contentSourceName** element of the request message.
- If the user profile import is in a state where it cannot be stopped or is already stopped, the protocol server MUST ignore the stop request.
- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.69.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.69.1.1 StopImportSoapIn

This is the request message for **StopImport** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **StopImport** element.

#### 3.1.4.69.1.2 StopImportSoapOut

This is a response message for **StopImport** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **StopImportResponse** element.

### 3.1.4.69.2 Elements

The following XML Schema element definitions are specific to this operation.

#### 3.1.4.69.2.1 StopImport

The definition of the **StopImport** element is as follows:

```
<s:element name="StopImport">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="contentSourceName" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**contentSourceName**: Name of the user profile content source for which to stop the import. This element MUST be present. MUST have one of the following values:
### Value | Meaning
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PEOPLE_IMPORT</td>
<td>Start the user profile import.</td>
</tr>
<tr>
<td>PEOPLE_DL_IMPORT</td>
<td>Start the distribution list import.</td>
</tr>
</tbody>
</table>

#### 3.1.4.69.2.2 StopImportResponse

The definition of the `StopImportResponse` element is as follows:

```xml
<s:element name="StopImportResponse">
    <s:complexType/>
</s:element>
```

#### 3.1.4.70 SystemDrive

This operation is not used and MUST NOT be invoked.

#### 3.1.4.71 TestCrawlRule

This operation is used to check if a specified URL matches the specified crawl rule.

```xml
<wsdl:operation name="TestCrawlRule">
    <wsdl:input message="TestCrawlRuleSoapIn然"
    <wsdl:output message="TestCrawlRuleSoapOut然"
</wsdl:operation>
```

The protocol client sends a `TestCrawlRuleSoapIn` request message and the protocol server responds with a `TestCrawlRuleSoapOut` response message, as follows:

- On receipt the protocol server validates the `versionIn` value from the request message. If it doesn't match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an `UpdatedConcurrencyException`.
- The protocol server MUST normalize the `path` element of the request message the same way it normalizes the `path` element of the `ActivateCrawlRule` operation request message.
- The protocol server MUST try to find the crawl rule in the crawl rules collection of the portal content project by the value of the normalized path. (The `path` element represents the crawl rule the protocol client wishes to test).
- If no matching crawl rule is found, the protocol server MUST throw a SOAP fault message containing a `DeletedConcurrencyException`.
- If a matching crawl rule is found, the protocol server first normalizes the URL specified by the `url` element of the request message the same way it normalizes the `path` element of the `ActivateCrawlRule` operation request message.
- The protocol server then matches the normalized URL with the crawl rule and MUST return true in the `TestCrawlRuleResult` element of the response message if the URL matches the crawl rule, and false in the `TestCrawlRuleResult` element of the response message if the URL does not match.
- On error, the protocol server MUST throw a SOAP fault message.
3.1.4.71.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.71.1.1 TestCrawlRuleSoapIn

This is the request message for the TestCrawlRule operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a TestCrawlRule element.

3.1.4.71.1.2 TestCrawlRuleSoapOut

This is a response message for the TestCrawlRule operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a TestCrawlRuleResponse element.

3.1.4.71.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.71.2.1 TestCrawlRule

TestCrawlRule element contains parameters for the TestCrawlRule operation.

```
<s:element name="TestCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="path" type="s:string" minOccurs="0"/>
      <s:element name="url" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

path: The path expression of the crawl rule that the protocol client wishes to test. MUST be present and the length MUST be greater than 0 and less than 2048 characters.

url: A URL to test against the crawl rule. MUST be present and the length MUST be greater than 0 and less than 2048 characters.

3.1.4.71.2.2 TestCrawlRuleResponse

The TestCrawlRuleResponse element contains the result of the TestCrawlRule operation.
3.1.4.72 TestCrawlRules

This operation is used to find the first crawl rule in the portal content project’s crawl rules collection that matches a specified URL.

```
<wsdl:operation name="TestCrawlRules">
  <wsdl:input message="TestCrawlRulesSoapIn"/>
  <wsdl:output message="TestCrawlRulesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **TestCrawlRulesSoapIn** request message and the protocol server responds with a **TestCrawlRulesSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.
- If there are no crawl rules in the portal content project’s crawl rules collection, the protocol server MUST return an empty string in the **TestCrawlRulesResult** element of the response message.
- Otherwise, the protocol server MUST first normalize the URL specified by the **url** element of the request message the same way it normalizes the **path** element of the **ActivateCrawlRule** operation request message.
- Then, the protocol server MUST check the normalized URL against all crawl rules in the collection and return the path expression of the first matching crawl rule.
- Otherwise, if the URL does not match any of the crawl rules the protocol server MUST return empty string in the **TestCrawlRulesResult** element of the response message.
- On error, the protocol server MUST throw a SOAP fault message.

3.1.4.72.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.72.1.1 TestCrawlRulesSoapIn

This is the request message for the **TestCrawlRules** operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a **TestCrawlRules** element.
3.1.4.72.1.2 TestCrawlRulesSoapOut
This is a response message for the TestCrawlRules operation.
The SOAP action value of the message is defined as:


The SOAP body contains a TestCrawlRulesResponse element.

3.1.4.72.2 Elements
The following XML Schema element definitions are specific to this operation.

3.1.4.72.2.1 TestCrawlRules
The TestCrawlRules element contains parameters for the TestCrawlRules operation.

<s:element name="TestCrawlRules">
    <s:complexType>
        <s:sequence>
            <s:element name="versionIn" type="s:int"/>
            <s:element name="url" type="s:string" minOccurs="0"/>
        </s:sequence>
    </s:complexType>
</s:element>

versionIn: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

url: A URL to test against the crawl rules. MUST be present and the length MUST be greater than 0 and less than 2048 characters.

3.1.4.72.2.2 TestCrawlRulesResponse
The TestCrawlRulesResponse element contains the result of the TestCrawlRules operation.

<s:element name="TestCrawlRulesResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="TestCrawlRulesResult" type="s:string" minOccurs="0"/>
        </s:sequence>
    </s:complexType>
</s:element>

TestCrawlRulesResult: The path expression of the crawl rule that matched the specified URL. This element MUST be present, and the value of this element MUST be empty string if there are no matching crawl rules.

3.1.4.73 UpdateCrawlRule
This operation is used to update a crawl rule for the portal content project.

<wsdl:operation name="UpdateCrawlRule">
    <wsdl:input message="UpdateCrawlRuleSoapIn"/>
    <wsdl:output message="UpdateCrawlRuleSoapOut"/>
The protocol client sends an **UpdateCrawlRuleSoapIn** request message and the protocol server responds with a **UpdateCrawlRuleSoapOut** response message, as follows:

- On receipt the protocol server validates the **versionIn** value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an **UpdatedConcurrencyException**.

- The protocol server then updates the current configuration version of the crawler application to the value of **versionIn+1**.

- Then, the protocol server MUST normalize the **path** parameter of the operation the same way it normalizes the **path** element of the **ActivateCrawlRule** operation request message. Then, if a crawl rule with the same normalized path does not exist in the portal content project, the protocol server MUST throw a SOAP fault message containing a **DeletedConcurrencyException**.

- Then, the protocol server MUST update the properties of the crawl rule using the elements of the **rule** parameter, setting the **path** (normalized the same way as the preceding path parameter), **type**, **suppressIndexing**, **followComplexUrls**, **crawlAsHttp**, **contentClass**, and **PluggableSecurityTrimmerId** properties of the **crawl rule** to the values of the corresponding elements of the **rule** parameter. Other elements of the **rule** parameter MUST be ignored.

On success, the protocol server MUST return the updated **path** property of the crawl rule in the **UpdateCrawlRuleResult** element of the response message.

- On error, the protocol server MUST throw a SOAP fault message.

### 3.1.4.73.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.73.1.1 UpdateCrawlRuleSoapIn

This is the request message for the **UpdateCrawlRule** operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains an **UpdateCrawlRule** element.

#### 3.1.4.73.1.2 UpdateCrawlRuleSoapOut

This is a response message for the **UpdateCrawlRule** operation.

The SOAP action value of the message is defined as:

```xml
```

The SOAP body contains an **UpdateCrawlRuleResponse** element.

### 3.1.4.73.2 Elements
The following XML Schema element definitions are specific to this operation.

### 3.1.4.73.2.1 UpdateCrawlRule

The **UpdateCrawlRule** element contains parameters for the **UpdateCrawlRule** operation.

```
<s:element name="UpdateCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="path" type="s:string" minOccurs="0"/>
      <s:element name="rule" type="tns:CrawlRuleInternal" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**versionIn**: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

**path**: The path expression of the crawl rule that needs to be updated. MUST be present and length MUST be greater than 0 and less than 2048 characters.

**rule**: The new values of the crawl rule properties. This element MUST be present.

### 3.1.4.73.2.2 UpdateCrawlRuleResponse

The **UpdateCrawlRuleResponse** element contains the result of **UpdateCrawlRule** operation.

```
<s:element name="UpdateCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateCrawlRuleResult" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**UpdateCrawlRuleResult**: The updated crawl rule path expression. MUST be present and length MUST be greater than 0 and less than 2048 characters.

### 3.1.4.74 ValidateApplicationPath

This operation is used to validate that a folder path is a viable location to host a full-text index catalog.

```
<wsdl:operation name="ValidateApplicationPath">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
  <wsdl:input message="ValidateApplicationPathSoapIn"/>
  <wsdl:output message="ValidateApplicationPathSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **ValidatePathSoapIn** request message and the protocol server responds with a **ValidateApplicationPathSoapOut** response message, as follows:

- If the format of the path element of the request message does not conform to the format of a valid folder path on the protocol server, the protocol server MUST throw SOAP fault message containing an **ArgumentException**.
If the path element is greater than 139 characters in length, or is a relative path, or does not specify a local drive on the protocol server, or specifies a local drive that is not present, or specifies the root folder on the local drive, or specifies removable media, the protocol server MUST throw SOAP fault message containing an ArgumentException.

If the local drive on the protocol server does not have at least the amount of free space, in bytes, as specified by the minFreeDiskSpace element of the request message, the protocol server MUST throw SOAP fault message containing an ArgumentException.

On success, the protocol server MUST send a ValidateApplicationPathResponse response with the value of the ValidateApplicationPathResult element set to the free drive space, in bytes, of the local drive on the protocol server for the folder path specified in the path element of the request message.

On success, the protocol server MUST return the free drive space, in bytes, of the local drive on the protocol server in the ValidateApplicationPathResult element of the response message.

On error, the protocol server MUST throw a SOAP fault message.

3.1.4.74.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.74.1.1 ValidateApplicationPathSoapIn

This is the request message for ValidateApplicationPath operation.

The SOAP action value of the message is defined as:


The SOAP body contains a ValidateApplicationPath element.

3.1.4.74.1.2 ValidateApplicationPathSoapOut

This is a response message for ValidateApplicationPath operation.

The SOAP action value of the message is defined as:


The SOAP body contains a ValidateApplicationPathResponse element.

3.1.4.74.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.74.2.1 ValidateApplicationPath

The definition of the ValidateApplicationPath element is as follows:

<s:element name="ValidateApplicationPath">
  <s:complexType>
    <s:sequence>
      <s:element name="path" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
path: The folder path on the protocol server. This element MUST be present. The value of this element MUST NOT be NULL and MUST NOT be empty. It MUST begin with a drive letter, and the length MUST be less than 140 characters.

minFreeDiskSpace: MUST be one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>The minimum free drive space required, in bytes.</td>
</tr>
<tr>
<td>Zero or Negative</td>
<td>Minimum free drive space required is zero byte.</td>
</tr>
</tbody>
</table>

3.1.4.74.2.2 ValidateApplicationPathResponse

The definition of the ValidateApplicationPathResponse element is as follows:

```
<element name="ValidateApplicationPathResponse">
  <complexType>
    <sequence>
      <element name="ValidateApplicationPathResult" type="s:long"/>
    </sequence>
  </complexType>
</element>
```

ValidateApplicationPathResult: Free disk drive space, in bytes, of the local drive on the protocol server containing the specified folder path. This value MUST be zero or positive.

3.1.4.75 ValidatePath

This operation is not used and MUST NOT be invoked.

3.1.4.76 ValidateScheduleTrigger

This operation is used to validate that a trigger, as specified in [MS-TSCH], section 2.4.2.11 Triggers, can be used to schedule a crawl.

```
<wsdl:operation name="ValidateScheduleTrigger">
  <wsdl:input message="ValidateScheduleTriggerSoapIn"/>
  <wsdl:output message="ValidateScheduleTriggerSoapOut"/>
</wsdl:operation>
```

The protocol client sends a ValidateScheduleTriggerSoapIn request message and the protocol server responds with a ValidateScheduleTriggerSoapOut response message, as follows:

- On receipt the protocol server validates the versionIn value from the request message. If it doesn’t match the current configuration version of the crawler application, the protocol server MUST throw a SOAP fault message containing an UpdatedConcurrencyException.
• Then, the protocol server tries to confirm that the trigger specified in the `trigger` element of the request message is a valid trigger for scheduling a crawl.
• On success, the protocol server MUST return details about the trigger in the `ValidateScheduleTriggerResult` element of the response message.
• On error, the protocol server MUST throw a SOAP fault message.

3.1.4.76.1 Messages
The following WSDL message definitions are specific to this operation.

3.1.4.76.1.1 ValidateScheduleTriggerSoapIn
This is the request message for `ValidateScheduleTrigger` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ValidateScheduleTrigger` element.

3.1.4.76.1.2 ValidateScheduleTriggerSoapOut
This is a response message for `ValidateScheduleTrigger` operation.

The SOAP action value of the message is defined as:

```
```

The SOAP body contains a `ValidateScheduleTriggerResponse` element.

3.1.4.76.2 Elements
The following XML Schema element definitions are specific to this operation.

3.1.4.76.2.1 ValidateScheduleTrigger
The definition of the `ValidateScheduleTrigger` element is as follows:

```
<s:element name="ValidateScheduleTrigger">
  <s:complexType>
    <s:sequence>
      <s:element name="versionIn" type="s:int"/>
      <s:element name="trigger" type="s:base64Binary" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

`versionIn`: The configuration version of the crawler application. This element MUST contain a value greater than or equal to zero.

`trigger`: A trigger, as specified in [MS-TSCH], section 2.4.2.11 Triggers to be validated. This element MUST be present and MUST be Base64 encoded.
3.1.4.76.2.2 ValidateScheduleTriggerResponse

The definition of the **ValidateScheduleTriggerResponse** element is as follows:

```xml
<s:element name="ValidateScheduleTriggerResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ValidateScheduleTriggerResult" type="tns:TriggerUpdate"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**ValidateScheduleTriggerResult**: The trigger details, as specified in [MS-TSCH], section 2.4.2.11 Triggers. This element MUST conform to the schema of the **TriggerUpdate** complex type.

3.1.4.76.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

3.1.4.76.3.1 TriggerUpdate

The definition of the **TriggerUpdate** type is as follows:

```xml
<s:complexType name="TriggerUpdate">
  <s:sequence>
    <s:element name="description" type="s:string" minOccurs="0"/>
    <s:element name="nextStart" type="s:dateTime"/>
  </s:sequence>
</s:complexType>
```

**description**: Description of the trigger, as specified in [MS-TSCH], section 2.4.2.11 Triggers. This element MUST be present if the trigger has a description.

**nextStart**: The next start time of the trigger, as specified in [MS-TSCH], section 2.4.2.11 Triggers.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.
4 Protocol Examples

4.1 Create and Configure a Content Source

Creating a content source typically involves the following messages:

1. Obtaining the current configuration version of the crawler application.
2. Retrieving the list of existing content sources.
3. Validation a crawl trigger.
4. Adding a new content source.
5. Updating the content source properties.

4.1.1 GetVersion

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<GetVersion
</soap:Body>
</soap:Envelope>
```

Response message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<GetVersionResponse
<GetVersionResult>170</GetVersionResult>
</GetVersionResponse>
</soap:Body>
</soap:Envelope>
```

4.1.2 GetContentSources

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<GetContentSources
</GetContentSources>
</soap:Body>
```

4.1.3 ValidateScheduleTrigger

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xm xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <soap:Body>
  <ValidateScheduleTrigger
    <versionIn>170</versionIn>
    <trigger>MAAAANgHAgAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA9</trigger>
  </ValidateScheduleTrigger>
 </soap:Body>
</soap:Envelope>
```
Response message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ValidateScheduleTriggerResponse
      <ValidateScheduleTriggerResult>
        <description>At 12:00 AM every day, starting 2/16/2008</description>
        <nextStart>2008-02-17T00:00:00</nextStart>
      </ValidateScheduleTriggerResult>
    </ValidateScheduleTriggerResponse>
  </soap:Body>
</soap:Envelope>
```

4.1.4 AddContentSource

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddContentSource
      <versionIn>170</versionIn>
      <catalog>0</catalog>
      <type>3</type>
      <name>Documents</name>
    </AddContentSource>
  </soap:Body>
</soap:Envelope>
```

Response message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddContentSourceResponse
      <AddContentSourceResult>12</AddContentSourceResult>
    </AddContentSourceResponse>
  </soap:Body>
</soap:Envelope>
```

4.1.5 EditContentSource

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <EditContentSource
    </EditContentSource>
  </soap:Body>
</soap:Envelope>
```
4.2 Create a Crawl Rule

Creating a crawl rule is illustrated by the following steps:

1. Getting the current list of crawl rules.
2. Creating a new crawl rule.
3. Updating properties of the new crawl rule.
4. Setting crawl account on the crawl rule.

4.2.1 GetCrawlRuleList

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    <GetCrawlRuleList versionIn>173</GetCrawlRuleList>
</soap:Envelope>
```
Response message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetCrawlRuleListResponse
      <GetCrawlRuleListResult />
    </GetCrawlRuleListResponse>
  </soap:Body>
</soap:Envelope>
```

### 4.2.2 AddCrawlRule

**Request message:**

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddCrawlRule
      <versionIn>173</versionIn>
      <isIncludeRule>true</isIncludeRule>
      <path>http://www.microsoft.com/*</path>
    </AddCrawlRule>
  </soap:Body>
</soap:Envelope>
```

**Response message:**

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddCrawlRuleResponse
      <AddCrawlRuleResult>http://www.microsoft.com/*</AddCrawlRuleResult>
    </AddCrawlRuleResponse>
  </soap:Body>
</soap:Envelope>
```

### 4.2.3 UpdateCrawlRule

**Request message:**

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <UpdateCrawlRule
      <versionIn>174</versionIn>
      <path>http://www.microsoft.com/*</path>
    </UpdateCrawlRule>
  </soap:Body>
</soap:Envelope>
```
4.2.4 SetCrawlRuleCredentials

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <soap:Body>
    <SetCrawlRuleCredentials xmlns="mws">
      <versionIn>175</versionIn>
      <path>http://www.microsoft.com/*</path>
      <authType>1</authType>
      <authString1>domainname\username</authString1>
      <authString2>password</authString2>
    </SetCrawlRuleCredentials>
  </soap:Body>
</soap:Envelope>
```

Response message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <soap:Body>
    <SetCrawlRuleCredentialsResponse xmlns="mws">
      <SetCrawlRuleCredentialsResult>176</SetCrawlRuleCredentialsResult>
    </SetCrawlRuleCredentialsResponse>
  </soap:Body>
</soap:Envelope>
```
4.3 Crawl a Content Source

Following is an example of the steps typically needed to begin a crawl:

1. Retrieving the list of existing content sources.
2. Requesting that a crawl be started.

4.3.1 GetContentSources

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetContentSources xmlns="http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService">
      <versionIn>180</versionIn>
      <catalog>0</catalog>
    </GetContentSources>
  </soap:Body>
</soap:Envelope>
```

Response message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetContentSourcesResponse xmlns="http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService">
      <GetContentSourcesResult>
        <contentSourcesInternal>
          <ContentSourceInternal>
            <id>4</id>
            <systemCreated>true</systemCreated>
            <type>1</type>
            <name>Local Office SharePoint Server sites</name>
            <wssCrawlStyle>0</wssCrawlStyle>
            <metadata />
            <followDirectories>false</followDirectories>
            <pageDepth>2147483647</pageDepth>
            <siteDepth>0</siteDepth>
            <startAddresses>
              <string>http://servername:500</string>
              <string>http://servername</string>
              <string>sps3://servername:500</string>
            </startAddresses>
            <throttleStart>0</throttleStart>
            <throttleDuration>0</throttleDuration>
            <nextIncCrawlStart>9999-12-31T23:59:59.9999999</nextIncCrawlStart>
            <nextFullCrawlStart>9999-12-31T23:59:59.9999999</nextFullCrawlStart>
            <dynamic>
              <crawlStatus>0</crawlStatus>
              <crawlStarted>2008-02-16T14:24:12</crawlStarted>
              <crawlCompleted>2008-02-16T14:24:32</crawlCompleted>
              <errorCount>0</errorCount>
              <crawlSuccesses>0</crawlSuccesses>
            </dynamic>
          </ContentSourceInternal>
          <ContentSourceInternal>
            <id>12</id>
```
4.3.2 StartCrawl

Request message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
      <versionIn>180</versionIn>
      <contentSource>12</contentSource>
      <type>0</type>
    </StartCrawl>
  </soap:Body>
</soap:Envelope>
```

Response message:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
      <StartCrawlResult>
        <crawlStatus>1</crawlStatus>
      </StartCrawlResult>
    </StartCrawlResponse>
  </soap:Body>
</soap:Envelope>
```
4.4 User Profile Import

Performing a user profile import typically involves the following steps:

1. Setting the user profile import domain and the user profile import account.
2. Requesting that the protocol server begin a user profile import.

4.4.1 SetImportAccount

Request message:

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SetImportAccount
      <versionIn>206</versionIn>
      <domain>domainname</domain>
      <userID>domainname\username</userID>
      <password>password</password>
    </SetImportAccount>
  </soap:Body>
</soap:Envelope>
```

Response message:

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SetImportAccountResponse
      <SetImportAccountResult>207</SetImportAccountResult>
    </SetImportAccountResponse>
  </soap:Body>
</soap:Envelope>
```

4.4.2 StartFullImport

Request message:

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
</soap:Body>
</soap:Envelope>
```
Response message:

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetContentSources xmlns="http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService">
      <versionIn>1</versionIn>
      <catalog>0</catalog>
    </GetContentSources>
  </soap:Body>
</soap:Envelope>

Response message:

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>The search configuration has been updated since it was last retrieved.</faultstring>
      <faultactor>http://servername:56737/SharedServices1/Search/SearchAdmin.asmx</faultactor>
      <detail>
        <exception xmlns="http://microsoft.com/webservices/OfficeServer/Search/SearchWebService">
          <!-- Additional details here -->
        </exception>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>

4.5 Failure Example

4.5.1 GetContentSources

Request message:

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetContentSources xmlns="http://microsoft.com/webservices/OfficeServer/Search/SearchApplicationWebService">
      <versionIn>1</versionIn>
      <catalog>0</catalog>
    </GetContentSources>
  </soap:Body>
</soap:Envelope>
The string value of the `detail.exception` element is the UTF-16 string representation of the following serialized exception:

```xml
  <SOAP-ENV:Body>
      <ClassName id="ref-3">Microsoft.Office.Server.Search.Administration.UpdatedConcurrencyException</ClassName>
      <Message id="ref-4">The search configuration has been updated since it was last retrieved.</Message>
      <Data xsi:null="1"/>
      <InnerException xsi:href="#ref-5"/>
    </StackTraceString>
    <RemoteStackTraceString xsi:null="1"/>
    <RemoteStackIndex>0</RemoteStackIndex>
    <ExceptionMethod id="ref-7">RunOnServer</ExceptionMethod>
    <HResult>-2146233088</HResult>
    <Source id="ref-8">Microsoft.Office.Server</Source>
  </a1:UpdatedConcurrencyException>
    <ClassName id="ref-9">System.Runtime.InteropServices.COMException</ClassName>
    <Message id="ref-10">An attempt was made to update application level configuration with an obsolete view of the current state.  Refresh your application-level objects and redo the update.</Message>
    <Data xsi:null="1"/>
    <InnerException xsi:null="1"/>
    <HelpURL xsi:null="1"/>
    </StackTraceString>
    <RemoteStackTraceString xsi:null="1"/>
    <RemoteStackIndex>0</RemoteStackIndex>
    <ExceptionMethod id="ref-12">ValidateVersionForRead</ExceptionMethod>
    <HResult>-2146233088</HResult>
    <Source id="ref-8">Microsoft.Office.Server</Source>
  </a3:COMException>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Microsoft.Office.Server.SearchAdministration.MSSITLB.IGatherApplication2
Void ValidateVersionForRead(Int32)
</ExceptionMethod>
<HRESULT>-2147217991</HRESULT>
<Source id="ref-13">
</Source>
</a3:COMException>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
5  Security

5.1  Security Considerations for Implementers

Because operations specified by this protocol modify configuration of the index server, in addition to the security considerations applicable to the underlying protocols, implementers are advised to consider user authorization on the server for performing the requested operation.

5.2  Index of Security Parameters

<table>
<thead>
<tr>
<th>Security Parameter</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
<td>section 2.1</td>
</tr>
</tbody>
</table>

6 Appendix A: Full WSDL

For ease of implementation the full WSDL is provided in the following code:

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
 xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/
 xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
 xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/
<wsdl:types>
<s:schema elementFormDefault="qualified"
  <s:element name="GetVersion">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetVersionResult" type="s:int"/>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetVersionResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetVersionResult" type="s:int"/>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="SetDefaultGatheringAccount">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
        <s:element minOccurs="0" maxOccurs="1" name="account" type="s:string"/>
        <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string"/>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="SetDefaultGatheringAccountResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="SetDefaultGatheringAccountResult" type="s:int"/>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetContentState">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetContentStateResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="GetContentStateResult" type="tns:ContentStateInternal"/>
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
</wsdl:types>
</wsdl:definitions>
```
<s:complexType name="ContentStateInternal">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="retryLimit" type="s:int"/>
    <s:element minOccurs="1" maxOccurs="1" name="lotusNotesConfigured" type="s:boolean"/>
    <s:element minOccurs="0" maxOccurs="1" name="defaultGatheringAccount" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="clientCertificateNames" type="tns:ArrayOfString"/>
    <s:element minOccurs="0" maxOccurs="1" name="activeDocuments" type="tns:ArrayOfString"/>
    <s:element minOccurs="0" maxOccurs="1" name="listKnownLotusNotesServers" type="tns:ArrayOfString"/>
  </s:sequence>
</s:complexType>

<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true" type="s:string"/>
  </s:sequence>
</s:complexType>

<s:element name="SetRetryLimit">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="retryLimit" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetRetryLimitResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="SetRetryLimitResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetContentSourcesMetadata">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="metadata" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetContentSourcesMetadataResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="SetContentSourcesMetadataResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="IndexSize">
  <s:complexType/>
</s:element>

<s:element name="IndexSizeResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="IndexSizeResult" type="s:long"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="CatalogPauseCrawlRequest">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="reason" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="CatalogPauseCrawlRequestResponse">
</s:element>

<s:element name="CatalogResumeCrawlRequest">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="reason" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="CatalogResumeCrawlRequestResponse">
</s:element>

<s:element name="CatalogPauseStatus">
  <s:complexType>
  </s:complexType>
</s:element>

<s:element name="CatalogPauseStatusResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="CatalogPauseStatusResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="IsCatalogPauseCompleted">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="reason" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="IsCatalogPauseCompletedResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="IsCatalogPauseCompletedResult" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="StartRankingUpdate">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="type" type="tns:RankingUpdateType"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="StartRankingUpdateResponse">
</s:element>

<s:simpleType name="RankingUpdateType">
  <s:restriction base="s:string">
    <s:enumeration value="FullUpdate"/>
  </s:restriction>
</s:simpleType>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="IsDeleteCrawlInProgressResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="IsDeleteCrawlInProgressResult" type="s:boolean"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="SearchDatabaseCleanup">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="SearchDatabaseCleanupResponse">
</s:complexType>
</s:element>
<s:element name="RefreshContentSource">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="id" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RefreshContentSourceResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="RefreshContentSourceResult" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RefreshAnchorContentSource">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RefreshAnchorContentSourceResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="RefreshAnchorContentSourceResult" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="serverName" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RemoveQueryServerResponse">
<s:complexType/>
</s:element>
<s:element name="StartCrawl">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="contentSource" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="type" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="StartCrawlResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="StartCrawlResult" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="PauseCrawl">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="contentSource" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="PauseCrawlResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="PauseCrawlResult" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="ResumeCrawl">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="contentSource" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="ResumeCrawlResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="ResumeCrawlResult" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="StopCrawl">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="StopCrawlResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="StopCrawlResult" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element minOccurs="1" maxOccurs="1" name="contentSource" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="StopCrawlResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="StopCrawlResult" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="ResetApp">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="ResetAppResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="ResetAppResult" type="s:int"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="MountApp">
<s:complexType/>
</s:element>
<s:element name="MountAppResponse">
<s:complexType/>
</s:element>
<s:element name="DismountApp">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="applicationName" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="DismountAppResponse">
<s:complexType/>
</s:element>
<s:element name="IsMounted">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="applicationName" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="IsMountedResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="IsMountedResult" type="s:boolean"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RemoveApp">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="applicationName" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RemoveAppResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="indexLocation" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
<s:sequence>
  <s:complexType>
    <s:element name="RemoveAppResponse">
      <s:complexType/>
    </s:element>
    <s:element name="GetGathererApplications">
      <s:complexType>
        <s:element name="GetGathererApplicationsResponse">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="GetGathererApplicationsResult" type="tns:ArrayOfString"/>
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="AddApp">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="role" type="tns:Role"/>
            </s:sequence>
            <s:simpleType name="Role">
              <s:restriction base="s:string">
                <s:enumeration value="None"/>
                <s:enumeration value="Index"/>
                <s:enumeration value="Query"/>
                <s:enumeration value="IndexQuery"/>
              </s:restriction>
            </s:simpleType>
            <s:element name="AddAppResponse">
              <s:complexType>
                <s:sequence>
                  <s:element minOccurs="1" maxOccurs="1" name="AddAppResult" type="s:boolean"/>
                </s:sequence>
              </s:complexType>
            </s:element>
            <s:element name="RemoveAllGathererApplications">
              <s:complexType>
                <s:element name="RemoveAllGathererApplicationsResponse">
                  <s:complexType>
                    <s:sequence>
                      <s:element minOccurs="1" maxOccurs="1" name="RemoveAllGathererApplicationsResult" type="s:int"/>
                    </s:sequence>
                  </s:complexType>
                </s:element>
              </s:element>
              <s:element name="ValidateScheduleTrigger">
                <s:complexType>
                  <s:sequence>
                    <s:element minOccurs="1" maxOccurs="1" name="ValidateScheduleTriggerResult" type="tns:TriggerUpdate"/>
                  </s:sequence>
                </s:complexType>
              </s:element>
              <s:element name="ValidateScheduleTriggerResponse">
                <s:complexType>
                  <s:sequence>
                    <s:element minOccurs="1" maxOccurs="1" name="ValidateScheduleTriggerResponseResult" type="tns:TriggerUpdate"/>
                  </s:sequence>
                </s:complexType>
              </s:element>
            </s:element>
          </s:complexType>
        </s:element>
      </s:element>
    </s:element>
  </s:complexType>
</s:sequence>
<s:complexType name="TriggerUpdate">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="description" type="s:string"/>
    <s:element minOccurs="1" maxOccurs="1" name="nextStart" type="s:dateTime"/>
  </s:sequence>
</s:complexType>

<s:element name="BackupRegistry">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="backupNow" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="BackupRegistryResponse">
  <s:complexType/>
</s:element>

<s:element name="AddExtension">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="ext" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="AddExtensionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="AddExtensionResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveExtension">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="ext" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveExtensionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RemoveExtensionResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ClearExtensionList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ClearExtensionListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ClearExtensionListResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetExtensionList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetExtensionListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetExtensionListResult" type="tns:ArrayOfString"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetIsExtensionIncludeList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="isInclude" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetIsExtensionIncludeListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="SetIsExtensionIncludeListResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="IsExtensionIncludeList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="IsExtensionIncludeListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="IsExtensionIncludeListResult" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="AddCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="isIncludeRule" type="s:boolean"/>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="AddCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="AddCrawlRuleResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RemoveCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RemoveCrawlRuleResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetCrawlRuleList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetCrawlRuleListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetCrawlRuleListResult" type="tns:ArrayOfCrawlRuleInternal"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:complexType name="ArrayOfCrawlRuleInternal">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="CrawlRuleInternal" nillable="true" type="tns:CrawlRuleInternal"/>
  </s:sequence>
</s:complexType>

<s:complexType name="CrawlRuleInternal">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    <s:element minOccurs="1" maxOccurs="1" name="type" type="s:int"/>
    <s:element minOccurs="1" maxOccurs="1" name="authenticationType" type="s:int"/>
    <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="contentClass" type="s:string"/>
    <s:element minOccurs="1" maxOccurs="1" name="suppressIndexing" type="s:boolean"/>
    <s:element minOccurs="1" maxOccurs="1" name="followComplexUrls" type="s:boolean"/>
    <s:element minOccurs="1" maxOccurs="1" name="crawlAsHttp" type="s:boolean"/>
    <s:element minOccurs="1" maxOccurs="1" name="enabled" type="s:boolean"/>
    <s:element minOccurs="0" maxOccurs="1" name="pluggableSecurityTrimmerId" type="s:int"/>
    <s:element minOccurs="0" maxOccurs="1" name="authUrl" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="authData" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="miscData" type="tns:ArrayOfString"/>
  </s:sequence>
</s:complexType>

<s:element name="UpdateCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="rule" type="tns:CrawlRuleInternal"/>
  </s:sequence>
</s:complexType>

<s:element name="UpdateCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="UpdateCrawlRuleResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ActivateCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ActivateCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ActivateCrawlRuleResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="TestCrawlRule">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="TestCrawlRuleResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="TestCrawlRuleResult" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="TestCrawlRules">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="TestCrawlRulesResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="TestCrawlRulesResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetCrawlRulePriority">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="rule" type="tns:CrawlRuleInternal"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ArrayofSitePathInternal">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="SitePathInternal" nillable="true" type="tns:SitePathInternal"/>
  </s:sequence>
</s:complexType>

<s:complexType name="SitePathInternal">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="contentClass" type="s:string"/>
  </s:sequence>
</s:complexType>

<s:element name="AddSiteRestriction">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="AddSiteRestrictionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="AddSiteRestrictionResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveSiteRestriction">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveSiteRestrictionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RemoveSiteRestrictionResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="AddSitePath">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="restrictionName" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="contentClass" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="AddSitePathResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="AddSitePathResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveSitePath">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveSitePathResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="RemoveSitePathResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType>
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
    <s:element minOccurs="0" maxOccurs="1" name="restrictionName" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
  </s:sequence>
</s:complexType>

<s:element name="RemoveSitePathResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RemoveSitePathResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ActivateProfileImport">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="enable" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ActivateProfileImportResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ActivateProfileImportResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetDefaultImportAccount">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="userID" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetDefaultImportAccountResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="SetDefaultImportAccountResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SetImportAccount">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="domain" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="userID" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="password" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="SetImportAccountResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="SetImportAccountResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetImportAccounts">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetImportAccountsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetImportAccountsResult" type="tns:ArrayOfImportDomainAccount"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:complexType name="ArrayOfImportDomainAccount">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="ImportDomainAccount" nillable="true" type="tns:ImportDomainAccount"/>
  </s:sequence>
</s:complexType>

<s:complexType name="ImportDomainAccount">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="domain" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="account" type="s:string"/>
  </s:sequence>
</s:complexType>

<s:element name="CleanUnreferencedSiteRestrictions">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="domains" type="tns:ArrayOfString"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="CleanUnreferencedSiteRestrictionsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="CleanUnreferencedSiteRestrictionsResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetContentSource">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="catalog" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="contentSourceName" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetContentSourceResponse">
  <s:complexType>
    <s:sequence>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="GetContentSourceResult" type="tns:ContentSourceInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:complexType name="ContentSourceInternal">
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="id" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="systemCreated" type="s:boolean"/>
<s:element minOccurs="1" maxOccurs="1" name="type" type="s:int"/>
<s:element minOccurs="0" maxOccurs="1" name="name" type="s:string"/>
<s:element minOccurs="1" maxOccurs="1" name="wssCrawlStyle" type="s:int"/>
<s:element minOccurs="0" maxOccurs="1" name="metadata" type="s:string"/>
<s:element minOccurs="1" maxOccurs="1" name="followDirectories" type="s:boolean"/>
<s:element minOccurs="1" maxOccurs="1" name="pageDepth" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="siteDepth" type="s:int"/>
<s:element minOccurs="0" maxOccurs="1" name="startAddresses" type="tns:ArrayOfString"/>
<s:element minOccurs="1" maxOccurs="1" name="throttleStart" type="s:int"/>
<s:element minOccurs="1" maxOccurs="1" name="throttleDuration" type="s:int"/>
<s:element minOccurs="0" maxOccurs="1" name="fullCrawlTrigger" type="s:base64Binary"/>
<s:element minOccurs="0" maxOccurs="1" name="incCrawlTrigger" type="s:base64Binary"/>
<s:element minOccurs="1" maxOccurs="1" name="nextIncCrawlStart" type="s:dateTime"/>
<s:element minOccurs="1" maxOccurs="1" name="nextFullCrawlStart" type="s:dateTime"/>
<s:element minOccurs="0" maxOccurs="1" name="incScheduleDescription" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name="fullScheduleDescription" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name="dynamic" type="tns:ContentSourceDynamicPropsInternal"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="StartFullImport">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
<s:element minOccurs="0" maxOccurs="1" name="contentSourceName" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="StartFullImportResponse">
</s:complexType>
</s:element>
<s:element name="StartIncrementalImport">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
<s:element minOccurs="0" maxOccurs="1" name="contentSourceName" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="StartIncrementalImportResponse">
</s:complexType>
</s:element>
<s:element name="StopImport">
<s:complexType>
</s:element>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
<s:complexType name="ArrayOfCrawlMappingInternal">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" nillable="true" type="tns:CrawlMappingInternal"/>
  </s:sequence>
</s:complexType>

<s:complexType name="CrawlMappingInternal">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="source" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="target" type="s:string"/>
  </s:sequence>
</s:complexType>

<s:element name="AddCrawlMapping">
  ...
</s:element>

<s:element name="AddContentSource">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="catalog" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="type" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="AddContentSourceResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="AddContentSourceResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveContentSource">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="catalog" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="id" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="RemoveContentSourceResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RemoveContentSourceResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetCrawlMappings">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="GetCrawlMappingsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetCrawlMappingsResult" type="tns:ArrayOfCrawlMappingInternal"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
  <s:element minOccurs="0" maxOccurs="1" name="source" type="s:string"/>
  <s:element minOccurs="0" maxOccurs="1" name="target" type="s:string"/>
</s:complexType>
</s:element>
<s:element name="AddCrawlMappingResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="AddCrawlMappingResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RemoveCrawlMapping">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="versionIn" type="s:int"/>
      <s:element minOccurs="0" maxOccurs="1" name="source" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RemoveCrawlMappingResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RemoveCrawlMappingResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ListKnownLotusNotesDatabases">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="serverName" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ListKnownLotusNotesDatabasesResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="ListKnownLotusNotesDatabasesResult" type="tns:ArrayOfString"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ValidateApplicationPath">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
      <s:element minOccurs="1" maxOccurs="1" name="minFreeDiskSpace" type="s:long"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ValidateApplicationPathResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ValidateApplicationPathResult" type="s:long"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ValidatePath">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
      <s:element minOccurs="1" maxOccurs="1" name="minFreeDiskSpace" type="s:long"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ValidatePathResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ValidatePathResult" type="s:long"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="DefaultIn
stallationIndexLocation">
  <s:complexType/>
</s:element>

<s:element name="DefaultInstallationIndexLocationResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="DefaultInstallationIndexLocationResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="SystemDrive">
  <s:complexType/>
</s:element>

<s:element name="SystemDriveResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="SystemDriveResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="IPAddress">
  <s:complexType/>
</s:element>

<s:element name="IPAddressResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="IPAddressResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ConfigurePropagationShare">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="account" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="ConfigurePropagationShareResponse">
  <s:complexType/>
</s:element>

<s:element name="IsPropagationShareConfigured">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="account" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<wsdl:message name="IsCatalogPauseCompletedSoapOut">
  <wsdl:part name="parameters" element="tns:IsCatalogPauseCompletedResponse"/>
</wsdl:message>

<wsdl:message name="StartRankingUpdateSoapIn">
  <wsdl:part name="parameters" element="tns:StartRankingUpdate"/>
</wsdl:message>

<wsdl:message name="StartRankingUpdateSoapOut">
  <wsdl:part name="parameters" element="tns:StartRankingUpdateResponse"/>
</wsdl:message>

<wsdl:message name="IsDeleteCrawlInProgressSoapIn">
  <wsdl:part name="parameters" element="tns:IsDeleteCrawlInProgress"/>
</wsdl:message>

<wsdl:message name="IsDeleteCrawlInProgressSoapOut">
  <wsdl:part name="parameters" element="tns:IsDeleteCrawlInProgressResponse"/>
</wsdl:message>

<wsdl:message name="SearchDatabaseCleanupSoapIn">
  <wsdl:part name="parameters" element="tns:SearchDatabaseCleanup"/>
</wsdl:message>

<wsdl:message name="SearchDatabaseCleanupSoapOut">
  <wsdl:part name="parameters" element="tns:SearchDatabaseCleanupResponse"/>
</wsdl:message>

<wsdl:message name="RefreshContentSourceSoapIn">
  <wsdl:part name="parameters" element="tns:RefreshContentSource"/>
</wsdl:message>

<wsdl:message name="RefreshContentSourceSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshContentSourceResponse"/>
</wsdl:message>

<wsdl:message name="RefreshAnchorContentSourceSoapIn">
  <wsdl:part name="parameters" element="tns:RefreshAnchorContentSource"/>
</wsdl:message>

<wsdl:message name="RefreshAnchorContentSourceSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshAnchorContentSourceResponse"/>
</wsdl:message>

<wsdl:message name="EditContentSourceSoapIn">
  <wsdl:part name="parameters" element="tns:EditContentSource"/>
</wsdl:message>

<wsdl:message name="EditContentSourceSoapOut">
  <wsdl:part name="parameters" element="tns:EditContentSourceResponse"/>
</wsdl:message>

<wsdl:message name="GetPropagationInternalSoapIn">
  <wsdl:part name="parameters" element="tns:GetPropagationInternal"/>
</wsdl:message>

<wsdl:message name="GetPropagationInternalSoapOut">
  <wsdl:part name="parameters" element="tns:GetPropagationInternalResponse"/>
</wsdl:message>

<wsdl:message name="RemoveQueryServerSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveQueryServer"/>
</wsdl:message>

<wsdl:message name="RemoveQueryServerSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveQueryServerResponse"/>
</wsdl:message>

<wsdl:message name="StartCrawlSoapIn">
  <wsdl:part name="parameters" element="tns:StartCrawl"/>
</wsdl:message>

<wsdl:message name="StartCrawlSoapOut">
  <wsdl:part name="parameters" element="tns:StartCrawlResponse"/>
</wsdl:message>

<wsdl:message name="PauseCrawlSoapIn">
  <wsdl:part name="parameters" element="tns:PauseCrawl"/>
</wsdl:message>

<wsdl:message name="PauseCrawlSoapOut">
  <wsdl:part name="parameters" element="tns:PauseCrawlResponse"/>
</wsdl:message>

<wsdl:message name="ResumeCrawlSoapIn">
  <wsdl:part name="parameters" element="tns:ResumeCrawl"/>
</wsdl:message>

<wsdl:message name="ResumeCrawlSoapOut">
  <wsdl:part name="parameters" element="tns:ResumeCrawlResponse"/>
</wsdl:message>
<wsdl:message name="StopCrawlSoapIn">
  <wsdl:part name="parameters" element="tns:StopCrawl"/>
</wsdl:message>
<wsdl:message name="StopCrawlSoapOut">
  <wsdl:part name="parameters" element="tns:StopCrawlResponse"/>
</wsdl:message>
<wsdl:message name="ResetAppSoapIn">
  <wsdl:part name="parameters" element="tns:ResetApp"/>
</wsdl:message>
<wsdl:message name="ResetAppSoapOut">
  <wsdl:part name="parameters" element="tns:ResetAppResponse"/>
</wsdl:message>
<wsdl:message name="MountAppSoapIn">
  <wsdl:part name="parameters" element="tns:MountApp"/>
</wsdl:message>
<wsdl:message name="MountAppSoapOut">
  <wsdl:part name="parameters" element="tns:MountAppResponse"/>
</wsdl:message>
<wsdl:message name="DismountAppSoapIn">
  <wsdl:part name="parameters" element="tns:DismountApp"/>
</wsdl:message>
<wsdl:message name="DismountAppSoapOut">
  <wsdl:part name="parameters" element="tns:DismountAppResponse"/>
</wsdl:message>
<wsdl:message name="IsMountedSoapIn">
  <wsdl:part name="parameters" element="tns:IsMounted"/>
</wsdl:message>
<wsdl:message name="IsMountedSoapOut">
  <wsdl:part name="parameters" element="tns:IsMountedResponse"/>
</wsdl:message>
<wsdl:message name="RemoveAppSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveApp"/>
</wsdl:message>
<wsdl:message name="RemoveAppSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveAppResponse"/>
</wsdl:message>
<wsdl:message name="GetGathererApplicationsSoapIn">
  <wsdl:part name="parameters" element="tns:GetGathererApplications"/>
</wsdl:message>
<wsdl:message name="GetGathererApplicationsSoapOut">
  <wsdl:part name="parameters" element="tns:GetGathererApplicationsResponse"/>
</wsdl:message>
<wsdl:message name="AddAppSoapIn">
  <wsdl:part name="parameters" element="tns:AddApp"/>
</wsdl:message>
<wsdl:message name="AddAppSoapOut">
  <wsdl:part name="parameters" element="tns:AddAppResponse"/>
</wsdl:message>
<wsdl:message name="RemoveAllGathererApplicationsSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveAllGathererApplications"/>
</wsdl:message>
<wsdl:message name="RemoveAllGathererApplicationsSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveAllGathererApplicationsResponse"/>
</wsdl:message>
<wsdl:message name="ValidateScheduleTriggerSoapIn">
  <wsdl:part name="parameters" element="tns:ValidateScheduleTrigger"/>
</wsdl:message>
<wsdl:message name="ValidateScheduleTriggerSoapOut">
  <wsdl:part name="parameters" element="tns:ValidateScheduleTriggerResponse"/>
</wsdl:message>
<wsdl:message name="BackupRegistrySoapIn">
  <wsdl:part name="parameters" element="tns:BackupRegistry"/>
</wsdl:message>
<wsdl:message name="BackupRegistrySoapOut">
  <wsdl:part name="parameters" element="tns:BackupRegistryResponse"/>
</wsdl:message>
<wsdl:message name="AddExtensionSoapIn">
  <wsdl:part name="parameters" element="tns:AddExtension"/>
</wsdl:message>
<wsdl:message name="AddExtensionSoapOut">
  <wsdl:part name="parameters" element="tns:AddExtensionResponse"/>
</wsdl:message>

<wsdl:message name="RemoveExtensionSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveExtension"/>
</wsdl:message>

<wsdl:message name="RemoveExtensionSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveExtensionResponse"/>
</wsdl:message>

<wsdl:message name="ClearExtensionListSoapIn">
  <wsdl:part name="parameters" element="tns:ClearExtensionList"/>
</wsdl:message>

<wsdl:message name="ClearExtensionListSoapOut">
  <wsdl:part name="parameters" element="tns:ClearExtensionListResponse"/>
</wsdl:message>

<wsdl:message name="GetExtensionListSoapIn">
  <wsdl:part name="parameters" element="tns:GetExtensionList"/>
</wsdl:message>

<wsdl:message name="GetExtensionListSoapOut">
  <wsdl:part name="parameters" element="tns:GetExtensionListResponse"/>
</wsdl:message>

<wsdl:message name="SetIsExtensionIncludeListSoapIn">
  <wsdl:part name="parameters" element="tns:SetIsExtensionIncludeList"/>
</wsdl:message>

<wsdl:message name="SetIsExtensionIncludeListSoapOut">
  <wsdl:part name="parameters" element="tns:SetIsExtensionIncludeListResponse"/>
</wsdl:message>

<wsdl:message name="IsExtensionIncludeListSoapIn">
  <wsdl:part name="parameters" element="tns:IsExtensionIncludeList"/>
</wsdl:message>

<wsdl:message name="IsExtensionIncludeListSoapOut">
  <wsdl:part name="parameters" element="tns:IsExtensionIncludeListResponse"/>
</wsdl:message>

<wsdl:message name="AddCrawlRuleSoapIn">
  <wsdl:part name="parameters" element="tns:AddCrawlRule"/>
</wsdl:message>

<wsdl:message name="AddCrawlRuleSoapOut">
  <wsdl:part name="parameters" element="tns:AddCrawlRuleResponse"/>
</wsdl:message>

<wsdl:message name="RemoveCrawlRuleSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveCrawlRule"/>
</wsdl:message>

<wsdl:message name="RemoveCrawlRuleSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveCrawlRuleResponse"/>
</wsdl:message>

<wsdl:message name="GetCrawlRuleListSoapIn">
  <wsdl:part name="parameters" element="tns:GetCrawlRuleList"/>
</wsdl:message>

<wsdl:message name="GetCrawlRuleListSoapOut">
  <wsdl:part name="parameters" element="tns:GetCrawlRuleListResponse"/>
</wsdl:message>

<wsdl:message name="UpdateCrawlRuleSoapIn">
  <wsdl:part name="parameters" element="tns:UpdateCrawlRule"/>
</wsdl:message>

<wsdl:message name="UpdateCrawlRuleSoapOut">
  <wsdl:part name="parameters" element="tns:UpdateCrawlRuleResponse"/>
</wsdl:message>

<wsdl:message name="ActivateCrawlRuleSoapIn">
  <wsdl:part name="parameters" element="tns:ActivateCrawlRule"/>
</wsdl:message>

<wsdl:message name="ActivateCrawlRuleSoapOut">
  <wsdl:part name="parameters" element="tns:ActivateCrawlRuleResponse"/>
</wsdl:message>

<wsdl:message name="TestCrawlRuleSoapIn">
  <wsdl:part name="parameters" element="tns:TestCrawlRule"/>
</wsdl:message>

<wsdl:message name="TestCrawlRuleSoapOut">
  <wsdl:part name="parameters" element="tns:TestCrawlRuleResponse"/>
</wsdl:message>
<wsdl:message name="TestCrawlRulesSoapIn">
  <wsdl:part name="parameters" element="tns:TestCrawlRules"/>
</wsdl:message>

<wsdl:message name="TestCrawlRulesSoapOut">
  <wsdl:part name="parameters" element="tns:TestCrawlRulesResponse"/>
</wsdl:message>

<wsdl:message name="SetCrawlRulePrioritySoapIn">
  <wsdl:part name="parameters" element="tns:SetCrawlRulePriority"/>
</wsdl:message>

<wsdl:message name="SetCrawlRulePrioritySoapOut">
  <wsdl:part name="parameters" element="tns:SetCrawlRulePriorityResponse"/>
</wsdl:message>

<wsdl:message name="SetCrawlRuleCredentialsSoapIn">
  <wsdl:part name="parameters" element="tns:SetCrawlRuleCredentials"/>
</wsdl:message>

<wsdl:message name="SetCrawlRuleCredentialsSoapOut">
  <wsdl:part name="parameters" element="tns:SetCrawlRuleCredentialsResponse"/>
</wsdl:message>

<wsdl:message name="GetSiteRestrictionListSoapIn">
  <wsdl:part name="parameters" element="tns:GetSiteRestrictionList"/>
</wsdl:message>

<wsdl:message name="GetSiteRestrictionListSoapOut">
  <wsdl:part name="parameters" element="tns:GetSiteRestrictionListResponse"/>
</wsdl:message>

<wsdl:message name="AddSiteRestrictionSoapIn">
  <wsdl:part name="parameters" element="tns:AddSiteRestriction"/>
</wsdl:message>

<wsdl:message name="AddSiteRestrictionSoapOut">
  <wsdl:part name="parameters" element="tns:AddSiteRestrictionResponse"/>
</wsdl:message>

<wsdl:message name="RemoveSiteRestrictionSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveSiteRestriction"/>
</wsdl:message>

<wsdl:message name="RemoveSiteRestrictionSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveSiteRestrictionResponse"/>
</wsdl:message>

<wsdl:message name="AddSitePathSoapIn">
  <wsdl:part name="parameters" element="tns:AddSitePath"/>
</wsdl:message>

<wsdl:message name="AddSitePathSoapOut">
  <wsdl:part name="parameters" element="tns:AddSitePathResponse"/>
</wsdl:message>

<wsdl:message name="RemoveSitePathSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveSitePath"/>
</wsdl:message>

<wsdl:message name="RemoveSitePathSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveSitePathResponse"/>
</wsdl:message>

<wsdl:message name="ActivateProfileImportSoapIn">
  <wsdl:part name="parameters" element="tns:ActivateProfileImport"/>
</wsdl:message>

<wsdl:message name="ActivateProfileImportSoapOut">
  <wsdl:part name="parameters" element="tns:ActivateProfileImportResponse"/>
</wsdl:message>

<wsdl:message name="SetDefaultImportAccountSoapIn">
  <wsdl:part name="parameters" element="tns:SetDefaultImportAccount"/>
</wsdl:message>

<wsdl:message name="SetDefaultImportAccountSoapOut">
  <wsdl:part name="parameters" element="tns:SetDefaultImportAccountResponse"/>
</wsdl:message>

<wsdl:message name="SetImportAccountSoapIn">
  <wsdl:part name="parameters" element="tns:SetImportAccount"/>
</wsdl:message>

<wsdl:message name="SetImportAccountSoapOut">
  <wsdl:part name="parameters" element="tns:SetImportAccountResponse"/>
</wsdl:message>

<wsdl:message name="GetImportAccountsSoapIn">
  <wsdl:part name="parameters" element="tns:GetImportAccounts"/>
</wsdl:message>
<wsdl:message name="GetImportAccountsSoapOut">
  <wsdl:part name="parameters" element="tns:GetImportAccountsResponse"/>
</wsdl:message>

<wsdl:message name="CleanUnreferencedSiteRestrictionsSoapIn">
  <wsdl:part name="parameters" element="tns:CleanUnreferencedSiteRestrictions"/>
</wsdl:message>

<wsdl:message name="CleanUnreferencedSiteRestrictionsSoapOut">
  <wsdl:part name="parameters" element="tns:CleanUnreferencedSiteRestrictionsResponse"/>
</wsdl:message>

<wsdl:message name="GetContentSourceSoapIn">
  <wsdl:part name="parameters" element="tns:GetContentSource"/>
</wsdl:message>

<wsdl:message name="GetContentSourceSoapOut">
  <wsdl:part name="parameters" element="tns:GetContentSourceResponse"/>
</wsdl:message>

<wsdl:message name="StartFullImportSoapIn">
  <wsdl:part name="parameters" element="tns:StartFullImport"/>
</wsdl:message>

<wsdl:message name="StartFullImportSoapOut">
  <wsdl:part name="parameters" element="tns:StartFullImportResponse"/>
</wsdl:message>

<wsdl:message name="StartIncrementalImportSoapIn">
  <wsdl:part name="parameters" element="tns:StartIncrementalImport"/>
</wsdl:message>

<wsdl:message name="StartIncrementalImportSoapOut">
  <wsdl:part name="parameters" element="tns:StartIncrementalImportResponse"/>
</wsdl:message>

<wsdl:message name="StopImportSoapIn">
  <wsdl:part name="parameters" element="tns:StopImport"/>
</wsdl:message>

<wsdl:message name="StopImportSoapOut">
  <wsdl:part name="parameters" element="tns:StopImportResponse"/>
</wsdl:message>

<wsdl:message name="GetImportStatusSoapIn">
  <wsdl:part name="parameters" element="tns:GetImportStatus"/>
</wsdl:message>

<wsdl:message name="GetImportStatusSoapOut">
  <wsdl:part name="parameters" element="tns:GetImportStatusResponse"/>
</wsdl:message>

<wsdl:message name="GetContentSourcesSoapIn">
  <wsdl:part name="parameters" element="tns:GetContentSources"/>
</wsdl:message>

<wsdl:message name="GetContentSourcesSoapOut">
  <wsdl:part name="parameters" element="tns:GetContentSourcesResponse"/>
</wsdl:message>

<wsdl:message name="AddContentSourceSoapIn">
  <wsdl:part name="parameters" element="tns:AddContentSource"/>
</wsdl:message>

<wsdl:message name="AddContentSourceSoapOut">
  <wsdl:part name="parameters" element="tns:AddContentSourceResponse"/>
</wsdl:message>

<wsdl:message name="RemoveContentSourceSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveContentSource"/>
</wsdl:message>

<wsdl:message name="RemoveContentSourceSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveContentSourceResponse"/>
</wsdl:message>

<wsdl:message name="GetCrawlMappingsSoapIn">
  <wsdl:part name="parameters" element="tns:GetCrawlMappings"/>
</wsdl:message>

<wsdl:message name="GetCrawlMappingsSoapOut">
  <wsdl:part name="parameters" element="tns:GetCrawlMappingsResponse"/>
</wsdl:message>

<wsdl:message name="AddCrawlMappingSoapIn">
  <wsdl:part name="parameters" element="tns:AddCrawlMapping"/>
</wsdl:message>

<wsdl:message name="AddCrawlMappingSoapOut">
  <wsdl:part name="parameters" element="tns:AddCrawlMappingResponse"/>
<wsdl:message name="RemoveCrawlMappingSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveCrawlMapping"/>
</wsdl:message>
<wsdl:message name="RemoveCrawlMappingSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveCrawlMappingResponse"/>
</wsdl:message>
<wsdl:message name="ListKnownLotusNotesDatabasesSoapIn">
  <wsdl:part name="parameters" element="tns:ListKnownLotusNotesDatabases"/>
</wsdl:message>
<wsdl:message name="ListKnownLotusNotesDatabasesSoapOut">
  <wsdl:part name="parameters" element="tns:ListKnownLotusNotesDatabasesResponse"/>
</wsdl:message>
<wsdl:message name="ValidateApplicationPathSoapIn">
  <wsdl:part name="parameters" element="tns:ValidateApplicationPath"/>
</wsdl:message>
<wsdl:message name="ValidateApplicationPathSoapOut">
  <wsdl:part name="parameters" element="tns:ValidateApplicationPathResponse"/>
</wsdl:message>
<wsdl:message name="ValidatePathSoapIn">
  <wsdl:part name="parameters" element="tns:ValidatePath"/>
</wsdl:message>
<wsdl:message name="ValidatePathSoapOut">
  <wsdl:part name="parameters" element="tns:ValidatePathResponse"/>
</wsdl:message>
<wsdl:message name="DefaultInstallationIndexLocationSoapIn">
  <wsdl:part name="parameters" element="tns:DefaultInstallationIndexLocation"/>
</wsdl:message>
<wsdl:message name="DefaultInstallationIndexLocationSoapOut">
  <wsdl:part name="parameters" element="tns:DefaultInstallationIndexLocationResponse"/>
</wsdl:message>
<wsdl:message name="SystemDriveSoapIn">
  <wsdl:part name="parameters" element="tns:SystemDrive"/>
</wsdl:message>
<wsdl:message name="SystemDriveSoapOut">
  <wsdl:part name="parameters" element="tns:SystemDriveResponse"/>
</wsdl:message>
<wsdl:message name="IPAddressSoapIn">
  <wsdl:part name="parameters" element="tns:IPAddress"/>
</wsdl:message>
<wsdl:message name="IPAddressSoapOut">
  <wsdl:part name="parameters" element="tns:IPAddressResponse"/>
</wsdl:message>
<wsdl:message name="ConfigurePropagationShareSoapIn">
  <wsdl:part name="parameters" element="tns:ConfigurePropagationShare"/>
</wsdl:message>
<wsdl:message name="ConfigurePropagationShareSoapOut">
  <wsdl:part name="parameters" element="tns:ConfigurePropagationShareResponse"/>
</wsdl:message>
<wsdl:message name="IsPropagationShareConfiguredSoapIn">
  <wsdl:part name="parameters" element="tns:IsPropagationShareConfigured"/>
</wsdl:message>
<wsdl:message name="IsPropagationShareConfiguredSoapOut">
  <wsdl:part name="parameters" element="tns:IsPropagationShareConfiguredResponse"/>
</wsdl:message>
<wsdl:portType name="SearchApplicationWebServiceSoap">
  <wsdl:operation name="GetVersion">
    <wsdl:input message="tns:GetVersionSoapIn"/>
    <wsdl:output message="tns:GetVersionSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SetDefaultGatheringAccount">
    <wsdl:input message="tns:SetDefaultGatheringAccountSoapIn"/>
    <wsdl:output message="tns:SetDefaultGatheringAccountSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetContentState">
    <wsdl:input message="tns:GetContentStateSoapIn"/>
    <wsdl:output message="tns:GetContentStateSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SetRetryLimit"/>
<wsdl:input message="tns:SetRetryLimitSoapIn"/>
<wsdl:output message="tns:SetRetryLimitSoapOut"/>
</wsdl:operation>

<wsdl:operation name="SetContentSourcesMetadata">
  <wsdl:input message="tns:SetContentSourcesMetadataSoapIn"/>
  <wsdl:output message="tns:SetContentSourcesMetadataSoapOut"/>
</wsdl:operation>

<wsdl:operation name="IndexSize">
  <wsdl:input message="tns:IndexSizeSoapIn"/>
  <wsdl:output message="tns:IndexSizeSoapOut"/>
</wsdl:operation>

<wsdl:operation name="CatalogPauseCrawlRequest">
  <wsdl:input message="tns:CatalogPauseCrawlRequestSoapIn"/>
  <wsdl:output message="tns:CatalogPauseCrawlRequestSoapOut"/>
</wsdl:operation>

<wsdl:operation name="CatalogResumeCrawlRequest">
  <wsdl:input message="tns:CatalogResumeCrawlRequestSoapIn"/>
  <wsdl:output message="tns:CatalogResumeCrawlRequestSoapOut"/>
</wsdl:operation>

<wsdl:operation name="CatalogPauseStatus">
  <wsdl:input message="tns:CatalogPauseStatusSoapIn"/>
  <wsdl:output message="tns:CatalogPauseStatusSoapOut"/>
</wsdl:operation>

<wsdl:operation name="IsCatalogPauseCompleted">
  <wsdl:input message="tns:IsCatalogPauseCompletedSoapIn"/>
  <wsdl:output message="tns:IsCatalogPauseCompletedSoapOut"/>
</wsdl:operation>

<wsdl:operation name="StartRankingUpdate">
  <wsdl:input message="tns:StartRankingUpdateSoapIn"/>
  <wsdl:output message="tns:StartRankingUpdateSoapOut"/>
</wsdl:operation>

<wsdl:operation name="IsDeleteCrawlInProgress">
  <wsdl:input message="tns:IsDeleteCrawlInProgressSoapIn"/>
  <wsdl:output message="tns:IsDeleteCrawlInProgressSoapOut"/>
</wsdl:operation>

<wsdl:operation name="SearchDatabaseCleanup">
  <wsdl:input message="tns:SearchDatabaseCleanupSoapIn"/>
  <wsdl:output message="tns:SearchDatabaseCleanupSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RefreshContentSource">
  <wsdl:input message="tns:RefreshContentSourceSoapIn"/>
  <wsdl:output message="tns:RefreshContentSourceSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RefreshAnchorContentSource">
  <wsdl:input message="tns:RefreshAnchorContentSourceSoapIn"/>
  <wsdl:output message="tns:RefreshAnchorContentSourceSoapOut"/>
</wsdl:operation>

<wsdl:operation name="EditContentSource">
  <wsdl:input message="tns:EditContentSourceSoapIn"/>
  <wsdl:output message="tns:EditContentSourceSoapOut"/>
</wsdl:operation>

<wsdl:operation name="GetPropagationInternal">
  <wsdl:input message="tns:GetPropagationInternalSoapIn"/>
  <wsdl:output message="tns:GetPropagationInternalSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RemoveQueryServer">
  <wsdl:input message="tns:RemoveQueryServerSoapIn"/>
  <wsdl:output message="tns:RemoveQueryServerSoapOut"/>
</wsdl:operation>

<wsdl:operation name="StartCrawl">
  <wsdl:input message="tns:StartCrawlSoapIn"/>
  <wsdl:output message="tns:StartCrawlSoapOut"/>
</wsdl:operation>

<wsdl:operation name="PauseCrawl">
  <wsdl:input message="tns:PauseCrawlSoapIn"/>
  <wsdl:output message="tns:PauseCrawlSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ResumeCrawl">
  <wsdl:input message="tns:ResumeCrawlSoapIn"/>
  <wsdl:output message="tns:ResumeCrawlSoapOut"/>
</wsdl:operation>
<wsdl:operation name="ResumeCrawl">
  <wsdl:input message="tns:ResumeCrawlSoapIn"/>
  <wsdl:output message="tns:ResumeCrawlSoapOut"/>
</wsdl:operation>

<wsdl:operation name="StopCrawl">
  <wsdl:input message="tns:StopCrawlSoapIn"/>
  <wsdl:output message="tns:StopCrawlSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ResetApp">
  <wsdl:input message="tns:ResetAppSoapIn"/>
  <wsdl:output message="tns:ResetAppSoapOut"/>
</wsdl:operation>

<wsdl:operation name="MountApp">
  <wsdl:input message="tns:MountAppSoapIn"/>
  <wsdl:output message="tns:MountAppSoapOut"/>
</wsdl:operation>

<wsdl:operation name="DismountApp">
  <wsdl:input message="tns:DismountAppSoapIn"/>
  <wsdl:output message="tns:DismountAppSoapOut"/>
</wsdl:operation>

<wsdl:operation name="IsMounted">
  <wsdl:input message="tns:IsMountedSoapIn"/>
  <wsdl:output message="tns:IsMountedSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RemoveApp">
  <wsdl:input message="tns:RemoveAppSoapIn"/>
  <wsdl:output message="tns:RemoveAppSoapOut"/>
</wsdl:operation>

<wsdl:operation name="GetGathererApplications">
  <wsdl:input message="tns:GetGathererApplicationsSoapIn"/>
  <wsdl:output message="tns:GetGathererApplicationsSoapOut"/>
</wsdl:operation>

<wsdl:operation name="AddApp">
  <wsdl:input message="tns:AddAppSoapIn"/>
  <wsdl:output message="tns:AddAppSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RemoveAllGathererApplications">
  <wsdl:input message="tns:RemoveAllGathererApplicationsSoapIn"/>
  <wsdl:output message="tns:RemoveAllGathererApplicationsSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ValidateScheduleTrigger">
  <wsdl:input message="tns:ValidateScheduleTriggerSoapIn"/>
  <wsdl:output message="tns:ValidateScheduleTriggerSoapOut"/>
</wsdl:operation>

<wsdl:operation name="BackupRegistry">
  <wsdl:input message="tns:BackupRegistrySoapIn"/>
  <wsdl:output message="tns:BackupRegistrySoapOut"/>
</wsdl:operation>

<wsdl:operation name="AddExtension">
  <wsdl:input message="tns:AddExtensionSoapIn"/>
  <wsdl:output message="tns:AddExtensionSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RemoveExtension">
  <wsdl:input message="tns:RemoveExtensionSoapIn"/>
  <wsdl:output message="tns:RemoveExtensionSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ClearExtensionList">
  <wsdl:input message="tns:ClearExtensionListSoapIn"/>
  <wsdl:output message="tns:ClearExtensionListSoapOut"/>
</wsdl:operation>

<wsdl:operation name="GetExtensionList">
  <wsdl:input message="tns:GetExtensionListSoapIn"/>
  <wsdl:output message="tns:GetExtensionListSoapOut"/>
</wsdl:operation>

<wsdl:operation name="SetIsExtensionIncludeList">
  <wsdl:input message="tns:SetIsExtensionIncludeListSoapIn"/>
  <wsdl:output message="tns:SetIsExtensionIncludeListSoapOut"/>
</wsdl:operation>

<wsdl:operation name="IsExtensionIncludeList">
  <wsdl:input message="tns:IsExtensionIncludeListSoapIn"/>
  <wsdl:output message="tns:IsExtensionIncludeListSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetImportAccounts">
  <wsdl:input message="tns:GetImportAccountsSoapIn"/>
  <wsdl:output message="tns:GetImportAccountsSoapOut"/>
</wsdl:operation>

<wsdl:operation name="CleanUnreferencedSiteRestrictions">
  <wsdl:input message="tns:CleanUnreferencedSiteRestrictionsSoapIn"/>
  <wsdl:output message="tns:CleanUnreferencedSiteRestrictionsSoapOut"/>
</wsdl:operation>

<wsdl:operation name="GetContentSource">
  <wsdl:input message="tns:GetContentSourceSoapIn"/>
  <wsdl:output message="tns:GetContentSourceSoapOut"/>
</wsdl:operation>

<wsdl:operation name="StartFullImport">
  <wsdl:input message="tns:StartFullImportSoapIn"/>
  <wsdl:output message="tns:StartFullImportSoapOut"/>
</wsdl:operation>

<wsdl:operation name="StartIncrementalImport">
  <wsdl:input message="tns:StartIncrementalImportSoapIn"/>
  <wsdl:output message="tns:StartIncrementalImportSoapOut"/>
</wsdl:operation>

<wsdl:operation name="StopImport">
  <wsdl:input message="tns:StopImportSoapIn"/>
  <wsdl:output message="tns:StopImportSoapOut"/>
</wsdl:operation>

<wsdl:operation name="GetImportStatus">
  <wsdl:input message="tns:GetImportStatusSoapIn"/>
  <wsdl:output message="tns:GetImportStatusSoapOut"/>
</wsdl:operation>

<wsdl:operation name="GetContentSources">
  <wsdl:input message="tns:GetContentSourcesSoapIn"/>
  <wsdl:output message="tns:GetContentSourcesSoapOut"/>
</wsdl:operation>

<wsdl:operation name="AddContentSource">
  <wsdl:input message="tns:AddContentSourceSoapIn"/>
  <wsdl:output message="tns:AddContentSourceSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RemoveContentSource">
  <wsdl:input message="tns:RemoveContentSourceSoapIn"/>
  <wsdl:output message="tns:RemoveContentSourceSoapOut"/>
</wsdl:operation>

<wsdl:operation name="GetCrawlMappings">
  <wsdl:input message="tns:GetCrawlMappingsSoapIn"/>
  <wsdl:output message="tns:GetCrawlMappingsSoapOut"/>
</wsdl:operation>

<wsdl:operation name="AddCrawlMapping">
  <wsdl:input message="tns:AddCrawlMappingSoapIn"/>
  <wsdl:output message="tns:AddCrawlMappingSoapOut"/>
</wsdl:operation>

<wsdl:operation name="RemoveCrawlMapping">
  <wsdl:input message="tns:RemoveCrawlMappingSoapIn"/>
  <wsdl:output message="tns:RemoveCrawlMappingSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ListKnownLotusNotesDatabases">
  <wsdl:input message="tns:ListKnownLotusNotesDatabasesSoapIn"/>
  <wsdl:output message="tns:ListKnownLotusNotesDatabasesSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ValidateApplicationPath">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Validates a file system path and a minimum amount of free disk space.</wsdl:documentation>
  <wsdl:input message="tns:ValidateApplicationPathSoapIn"/>
  <wsdl:output message="tns:ValidateApplicationPathSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ValidatePath">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Validates a file system path and a minimum amount of free disk space.</wsdl:documentation>
  <wsdl:input message="tns:ValidatePathSoapIn"/>
  <wsdl:output message="tns:ValidatePathSoapOut"/>
</wsdl:operation>

<wsdl:operation name="DefaultInstallationIndexLocation"/>
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the default installation index location, that is, the path to the Data folder under the application installation root.</wsdl:documentation>
<wsdl:input message="tns:DefaultInstallationIndexLocationSoapIn"/>
<wsdl:output message="tns:DefaultInstallationIndexLocationSoapOut"/>
</wsdl:operation>

<wsdl:operation name="SystemDrive">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the system drive letter.</wsdl:documentation>
<wsdl:input message="tns:SystemDriveSoapIn"/>
<wsdl:output message="tns:SystemDriveSoapOut"/>
</wsdl:operation>

<wsdl:operation name="IPAddress">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the static IP address of this server if one exists. Otherwise, the dynamic IP address.</wsdl:documentation>
<wsdl:input message="tns:IPAddressSoapIn"/>
<wsdl:output message="tns:IPAddressSoapOut"/>
</wsdl:operation>

<wsdl:operation name="ConfigurePropagationShare">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Ensures that the propagation share with the appropriate ACL.</wsdl:documentation>
<wsdl:input message="tns:ConfigurePropagationShareSoapIn"/>
<wsdl:output message="tns:ConfigurePropagationShareSoapOut"/>
</wsdl:operation>

<wsdl:operation name="IsPropagationShareConfigured">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Checks whether the propagation share exists and whether it is configured with the appropriate ACL.</wsdl:documentation>
<wsdl:input message="tns:IsPropagationShareConfiguredSoapIn"/>
<wsdl:output message="tns:IsPropagationShareConfiguredSoapOut"/>
</wsdl:operation>
</wsdl:portType>

<wsdl:binding name="SearchApplicationWebServiceSoap" type="tns:SearchApplicationWebServiceSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="GetVersion">
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetDefaultGatheringAccount">
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetContentState">
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetVersion">
<wsdl:operation name="SetRetryLimit">
  <soap:operation
      SetRetryLimit" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetContentSourcesMetadata">
  <soap:operation
      SetContentSourcesMetadata" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="IndexSize">
  <soap:operation
      IndexSize" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="CatalogPauseCrawlRequest">
  <soap:operation
      CatalogPauseCrawlRequest" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="CatalogResumeCrawlRequest">
  <soap:operation
      CatalogResumeCrawlRequest" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="CatalogPauseStatus">
  <soap:operation
      CatalogPauseStatus" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="IsCatalogPauseCompleted">
  <soap:operation
      IsCatalogPauseCompleted" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<soap:operation
    IsCatalogPauseCompleted" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap:operation
    StartRankingUpdate" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap:operation
    IsDeleteCrawlInProgress" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap:operation
    SearchDatabaseCleanup" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap:operation
    RefreshContentSource" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap:operation
    RefreshAnchorContentSource" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap:operation
    EditContentSource" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPropagationInternal">
  <soap:operation
    GetPropagationInternal" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveQueryServer">
  <soap:operation
    RemoveQueryServer" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="StartCrawl">
  <soap:operation
    StartCrawl" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="PauseCrawl">
  <soap:operation
    PauseCrawl" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ResumeCrawl">
  <soap:operation
    ResumeCrawl" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="StopCrawl">
  <soap:operation
    StopCrawl" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ResetApp">
  <soap:operation
    ResetApp" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="MountApp">
  <soap:operation
    MountApp" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DismountApp">
  <soap:operation
    DismountApp" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="IsMounted">
  <soap:operation
    IsMounted" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveApp">
  <soap:operation
    RemoveApp" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetGathererApplications">
  <soap:operation
    GetGathererApplications" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddApp">
<soap:operation
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveAllGathererApplications">
<soap:operation
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="ValidateScheduleTrigger">
<soap:operation
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="BackupRegistry">
<soap:operation
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="AddExtension">
<soap:operation
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveExtension">
<soap:operation
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="ClearExtensionList"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="UpdateCrawlRule">
  <soap:operation
    UpdateCrawlRule" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="ActivateCrawlRule">
  <soap:operation
    ActivateCrawlRule" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="TestCrawlRule">
  <soap:operation
    TestCrawlRule" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="TestCrawlRules">
  <soap:operation
    TestCrawlRules" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetCrawlRulePriority">
  <soap:operation
    SetCrawlRulePriority" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetCrawlRuleCredentials">
  <soap:operation
    SetCrawlRuleCredentials" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>

<wsdl:operation name="GetSiteRestrictionList">
  <soap:operation
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="AddSiteRestriction">
  <soap:operation
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveSiteRestriction">
  <soap:operation
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="AddSitePath">
  <soap:operation
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveSitePath">
  <soap:operation
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="ActivateProfileImport">
  <soap:operation
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetDefaultImportAccount">
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetImportAccount">
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetImportAccounts">
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="CleanUnreferencedSiteRestrictions">
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetContentSource">
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="StartFullImport">
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="StartIncrementalImport">
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<soap:operation
StartIncrementalImport" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="StopImport"
StopImport" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetImportStatus"
GetImportStatus" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetContentSources"
GetContentSources" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddContentSource"
AddContentSource" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveContentSource"
RemoveContentSource" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCrawlMappings"
GetCrawlMappings" style="document"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="AddCrawlMapping">
  <soap:operation
    AddCrawlMapping" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveCrawlMapping">
  <soap:operation
    RemoveCrawlMapping" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="ListKnownLotusNotesDatabases">
  <soap:operation
    ListKnownLotusNotesDatabases" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="ValidateApplicationPath">
  <soap:operation
    ValidateApplicationPath" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="ValidatePath">
  <soap:operation
    ValidatePath" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="DefaultInstallationIndexLocation">
  <soap:operation
    DefaultInstallationIndexLocation" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetContentState">
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetRetryLimit">
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetContentSourcesMetadata">
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="IndexSize">
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="CatalogPauseCrawlRequest">
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="CatalogResumeCrawlRequest">
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
<wsdl:operation name="CatalogPauseStatus">
  <soap12:operation
    CatalogPauseStatus" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="IsCatalogPauseCompleted">
  <soap12:operation
    IsCatalogPauseCompleted" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="StartRankingUpdate">
  <soap12:operation
    StartRankingUpdate" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="IsDeleteCrawlInProgress">
  <soap12:operation
    IsDeleteCrawlInProgress" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="SearchDatabaseCleanup">
  <soap12:operation
    SearchDatabaseCleanup" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RefreshContentSource">
  <soap12:operation
    RefreshContentSource" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RefreshAnchorContentSource">
<soap12:operation
    RefreshAnchorContentSource" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    EditContentSource" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    GetPropagationInternal" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    RemoveQueryServer" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    StartCrawl" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    PauseCrawl" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    ResumeCrawl" style="document"/>
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>

<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>

<wsdl:operation name="StopCrawl">
  <soap12:operation
    StopCrawl" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="ResetApp">
  <soap12:operation
    ResetApp" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="MountApp">
  <soap12:operation
    MountApp" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="DismountApp">
  <soap12:operation
    DismountApp" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="IsMounted">
  <soap12:operation
    IsMounted" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveApp">
  <soap12:operation
    RemoveApp" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
</wsdl:operation>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>

<wSDL:operation>
  <wsdl:operation name="GetGathererApplications">
    <soap12:operation
      GetGathererApplications" style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="AddApp">
    <soap12:operation
      AddApp" style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RemoveAllGathererApplications">
    <soap12:operation
      RemoveAllGathererApplications" style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ValidateScheduleTrigger">
    <soap12:operation
      ValidateScheduleTrigger" style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BackupRegistry">
    <soap12:operation
      BackupRegistry" style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="AddExtension">
    <soap12:operation
      AddExtension" style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:operation>
<wsdl:operation name="RemoveExtension">
  <soap12:operation
    RemoveExtension" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="ClearExtensionList">
  <soap12:operation
    ClearExtensionList" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetExtensionList">
  <soap12:operation
    GetExtensionList" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetIsExtensionIncludeList">
  <soap12:operation
    SetIsExtensionIncludeList" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="IsExtensionIncludeList">
  <soap12:operation
    IsExtensionIncludeList" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="AddCrawlRule">
  <soap12:operation
    AddCrawlRule" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveCrawlRule">
<soap12:operation
    RemoveCrawlRule" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    GetCrawlRuleList" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    UpdateCrawlRule" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    ActivateCrawlRule" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    TestCrawlRule" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    TestCrawlRules" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>

<soap12:operation
    SetCrawlRulePriority" style="document"/>
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
<wsdl:operation name="SetCrawlRuleCredentials">
  <soap12:operation
    SetCrawlRuleCredentials" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSiteRestrictionList">
  <soap12:operation
    GetSiteRestrictionList" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddSiteRestriction">
  <soap12:operation
    AddSiteRestriction" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveSiteRestriction">
  <soap12:operation
    RemoveSiteRestriction" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddSitePath">
  <soap12:operation
    AddSitePath" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveSitePath">
  <soap12:operation
    RemoveSitePath" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ActivateProfileImport">
  <soap12:operation
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetDefaultImportAccount">
  <soap12:operation
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetImportAccount">
  <soap12:operation
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetImportAccounts">
  <soap12:operation
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CleanUnreferencedSiteRestrictions">
  <soap12:operation
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetContentSource">
  <soap12:operation
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="StartFullImport">
  <soap12:operation
      StartFullImport" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="StartIncrementalImport">
  <soap12:operation
      StartIncrementalImport" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="StopImport">
  <soap12:operation
      StopImport" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetImportStatus">
  <soap12:operation
      GetImportStatus" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetContentSources">
  <soap12:operation
      GetContentSources" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="AddContentSource">
  <soap12:operation
      AddContentSource" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RemoveContentSource">
<soap12:operation
RemoveContentSource" style="document"/>
</wsdl:input>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCrawlMappings">
<soap12:operation
GetCrawlMappings" style="document"/>
</wsdl:input>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddCrawlMapping">
<soap12:operation
AddCrawlMapping" style="document"/>
</wsdl:input>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveCrawlMapping">
<soap12:operation
RemoveCrawlMapping" style="document"/>
</wsdl:input>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ListKnownLotusNotesDatabases">
<soap12:operation
ListKnownLotusNotesDatabases" style="document"/>
</wsdl:input>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ValidateApplicationPath">
<soap12:operation
ValidateApplicationPath" style="document"/>
</wsdl:input>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ValidatePath">
<soap12:operation
ValidatePath" style="document"/>
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="DefaultInstallationIndexLocation">
  <soap12:operation
      DefaultInstallationIndexLocation" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SystemDrive">
  <soap12:operation
      SystemDrive" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="IPAddress">
  <soap12:operation
      IPAddress" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ConfigurePropagationShare">
  <soap12:operation
      ConfigurePropagationShare" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="IsPropagationShareConfigured">
  <soap12:operation
      IsPropagationShareConfigured" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Office SharePoint Server 2007

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 3.1.4: This functionality was added as part of the Office SharePoint Server 2007 Infrastructure Update.

<2> Section 3.1.4: This functionality was added as part of the Office SharePoint Server 2007 Infrastructure Update.
8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class Major means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class Minor means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class None means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.1.5 Crawl Rule</td>
<td>Update lower case use of normative term in description for Crawl Rule.</td>
<td>Major</td>
</tr>
<tr>
<td>3.1.4.19.2.1 EditContentSource</td>
<td>Update lower case use of normative term in description for wssCrawStyle values.</td>
<td>Major</td>
</tr>
</tbody>
</table>
9 Index

A

Abstract data model
server 32
AddContentSource example 149
AddCrawlRule example 151
Anchor content source
server 38
Applicability 20
ArgumentException exception type 30
ArrayOfString complex type 22
Attribute groups 28
Attributes 28

C

Capability negotiation 20
Change tracking 219
COMException exception type 29
Complex types 22
ArrayOfString 22
ContentSourceDynamicPropsInternal 22
ContentSourceInternal 26
CrawlRuleInternal 23
ConcurrencyException exception type 28
Content source
server 35
ContentSourceDynamicPropsInternal complex type
22
ContentSourceInternal complex type 26
Crawl a content source example 153
Crawl mapping
server 38
Crawl rule
server 36
Crawler application
server 33
CrawlRuleInternal complex type 23
Create a crawl rule example 150
Create and configure a content source example 147

D

Data model - abstract
server 32
DeletedConcurrencyException exception type 29

E

EditContentSource example 149
Events
local - server 146
timer - server 146
Examples
AddContentSource 149
AddCrawlRule 151
crawl a content source 153
create a crawl rule 150
create and configure a content source 147
EditContentSource 149

GetContentSource (section 4.1.2 147, section 4.5.1 156)
GetContentSources 153
GetCrawlRuleList 150
GetVersion 147
SetCrawlRuleCredentials 152
SetImportAccount 155
StartCrawl 154
StartFullImport 155
UpdateCrawlRule 151
user profile import 155
ValidateScheduleTrigger 148

Exception types
ArgumentException 30
COMException 29
ConcurrencyException 28
DeletedConcurrencyException 29
ExternalException 29
InvalidEnumArgumentException 30
InvalidOperationException 30
KeyNotFoundException 30
UpdatedConcurrencyException 28
ExternalException exception type 29

F

Fields - vendor-extensible 20
Full WSDL 161

G

GetContentSource example (section 4.1.2 147,
section 4.5.1 156)
GetContentSources example 153
GetCrawlRuleList example 150
GetVersion example 147
Glossary 14
Groups 28

I

Implementer - security considerations 160
Index of security parameters 160
Informative references 19
Initialization
server 39
Introduction 14
InvalidEnumArgumentException exception type 30
InvalidOperationException exception types 30

K

KeyNotFoundException exception type 30

L

Local events
server 146

M
Message processing

server 40

Messages
ArrayOfString complex type 22
attribute groups 28
attributes 28
complex types 22
ContentSourceDynamicPropsInternal complex type 22
ContentSourceInternal complex type 26
CrawlRuleInternal complex type 23
elements 22
enumerated 22
exception types 28
groups 28
namespaces 21
simple types 28
syntax 21
transport 21

N

Namespaces 21
Normative references 18

O

Object hierarchy
server 32

Operations
ActivateCrawlRule 42
ActivateProfileImport 44
AddApp 46
AddContentSource 46
AddCrawlMapping 48
AddCrawlRule 50
AddExtension 52
AddSitePath 54
AddSiteRestriction 54
BackupRegistry 54
CatalogPauseCrawlRequest 56
CatalogPauseStatus 57
CatalogResumeCrawlRequest 59
CleanUnreferencedSiteRestrictions 60
ClearExtensionList 62
ConfigurePropagationShare 63
DefaultInstallationIndexLocation 63
DismountApp 64
EditContentSource 65
GetContentSource 68
GetContentSources 70
GetContentState 72
GetCrawlMappings 75
GetCrawlRuleList 77
GetExtensionList 78
GetGathererApplications 80
GetImportAccounts 80
GetImportStatus 82
GetPropagationInternal 84
GetSiteRestrictionList 84
GetVersion 84
IndexSize 86
IPAddress 87
IsCatalogPauseCompleted 87
IsDeleteCrawlInProgress 89
IsExtensionIncludeList 90
IsMounted 91
IsPropagationShareConfigured 91
ListKnownLotusNotesDatabases 92
MountApp 93
PauseCrawl 93
RefreshAnchorContentSource 95
RefreshContentSource 96
RemoveAllGathererApplications 98
RemoveApp 98
RemoveContentSource 100
RemoveCrawlMapping 101
RemoveCrawlRule 103
RemoveExtension 105
RemoveQueryServer 106
RemoveSitePath 106
RemoveSiteRestriction 106
ResetApp 107
ResumeCrawl 108
SearchDatabaseCleanup 110
SetContentSourcesMetadata 111
SetColorRuleCredentials 113
SetCrawlRulePriority 117
SetDefaultGatheringAccount 119
SetDefaultImportAccount 120
SetImportAccount 122
SetIsExtensionIncludeList 124
SetRetryLimit 126
StartCrawl 127
StartFullImport 129
StartIncrementalImport 131
StartRankingUpdate 132
StopCrawl 134
StopImport 135
SystemDrive 137
TestCrawlRule 137
TestCrawlRules 139
UpdateCrawlRule 140
ValidateApplicationPath 142
ValidatePath 144
ValidateScheduleTrigger 144
Overview (synopsis) 19

P

Parameters - security index 160
Portal content project
server 34

Preconditions 20
Prerequisites 20
Product behavior 218

Protocol Details
overview 32

R

References 18
informative 19
normative 18

Relationship to other protocols 20

S

Security
implementer considerations 160