



Using SFX in a Consortium Environment

Version 4

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris Bridge to Knowledge , Primo, Aleph, Voyager, SFX, MetaLib, Verde, DigiTool, Rosetta, bX, URM, Alma , and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Table of Contents

Chapter 1	Consortia Models	5
	Model 1: Multiple SFX Instances - Institutional Autonomy.....	5
	Model 2: Multiple SFX Instances - Including One Instance for Shared Resources	6
	Model 3: Single SFX Instance - Multiple Institutions Sharing the Same Instance	9
Chapter 2	Setting Up Model 2: Using a Shared SFX Instance	11
	Overview.....	11
	General Set Up	12
	Using the IP Institute Table to Handle Requests Sent to a Shared Instance	18
	Statistics in Model 2.....	23
	A-Z List and Institutes Using Consortium API.....	24
	Rapid Service Indicator Index and Consortia API.....	24
	Elsevier ScienceDirect Autoloader and Consortia API.....	26
Chapter 3	Multiple Institutions Sharing One Instance	27
	Overview.....	27
	IP Institute and Institute Thresholds	27
	Support for Multiple Proxies	28
	Storing Institute-Specific Linking Parameters for a Particular Target	28
	Creating HTML e-Journal Lists	31
	<i>Creating Complete e-Journal Lists</i>	31
	<i>Creating Institute-Specific e-Journal Lists</i>	32
	Configuring the Elsevier ScienceDirect Autoloader.....	32
Chapter 4	Locator.cgi	35
	Overview.....	35
	General Set Up	35
	Locating the Appropriate SFX Instance Based on IP Range.....	37
	Locating the Appropriate SFX Instance Based on a Cookie.....	37

Appendix A	Set-Up for Different Model 2 Scenarios	39
	Scenario 1 – Requests Begin at Local Instance	39
	Scenario 2 – Bi-Directional API	40
	Scenario 3 – Chain of Remote Instances	41
	Scenario 4 – Consecutive Remote Instances	42
	Scenario 5 – Instance Specific Request Forwarding	42
Appendix B	eBook Search Set-Up for Consortium Model 2	43
	Overview	43
	Configuration for Consortia API - eBook Search.....	44
	<i>Configuration in the Local Instance Where Local and Shared Instances Exist in the Same SFX Installation</i>	44
	<i>Configuration in a Shared Instance Where Local and Shared Instances Exist in the Same SFX Installation</i>	45
	<i>Configuration in the Local Instance Where Local and Shared Instances Do Not Exist in the Same SFX Installation</i>	45
	<i>Configuration in a Shared Instance Where Local and Shared Instances Do Not Exist in the Same SFX Installation</i>	46
	Setting Up Replication for RSI in Both the Shared (Master) and Local (Slave) Instances.....	47
	Rebuilding A-Z eBooks in the Remote Shared Instance	48
	Troubleshooting and Debugging	49

1

Consortia Models

This section includes:

- **Model 1: Multiple SFX Instances - Institutional Autonomy** on page 5
- **Model 2: Multiple SFX Instances - Including One Instance for Shared Resources** on page 6
- **Model 3: Single SFX Instance - Multiple Institutions Sharing the Same Instance** on page 9

Model 1: Multiple SFX Instances - Institutional Autonomy

This model is characterized by the following:

- Each local institution has its own instance and is responsible for all data and configuration.
- All SFX instances are part of the same SFX installation and benefit from shared administration tasks such as back-up.
- Implementation of this model is typically determined in cases where there is little or no sharing of resources and services and/or where there is little central administration.



Figure 1: Multiple SFX Instances - Institutional Autonomy

Each local instance has total control over the design and layout of their SFX menu.

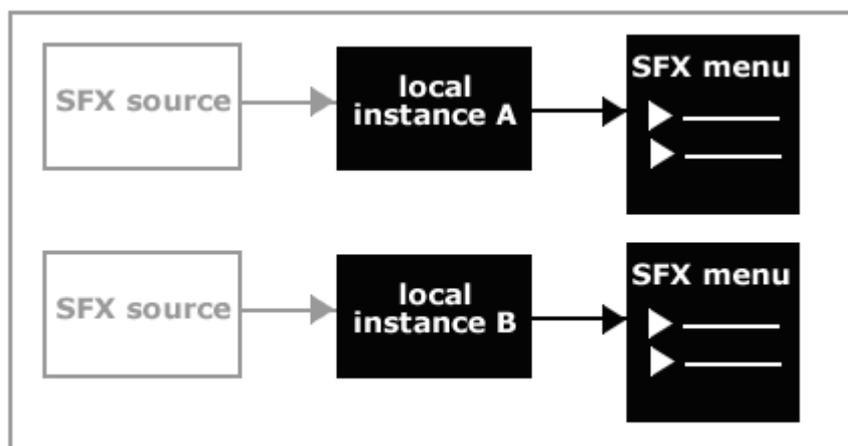


Figure 2: Total Control over SFX Menu

Model 2: Multiple SFX Instances - Including One Instance for Shared Resources

This model is characterized by the following:

- Implementation of this model depends on a central office being responsible for all shared data.
- This model requires less time from each local instance to implement and maintain.

- Each local institution has its own instance and is responsible for its non-shared data.
- A central office manages the shared resources instance and is responsible for all shared data.
- Interaction between the SFX instances results in the generation of an SFX menu screen.



Figure 3: Multiple SFX Instances - Including One Instance for Shared Resources

- **Scenario 1** - Each local instance maintains its own SFX sources. For each of these sources, the local instance has control over the design and layout of the SFX menu.

First, an SFX source sends an OpenURL to a local SFX instance. The instance forwards this OpenURL to the shared resources instance and receives information about services. After receiving this information, the local instance determines which services, both local and remote, to present in the SFX menu screen.

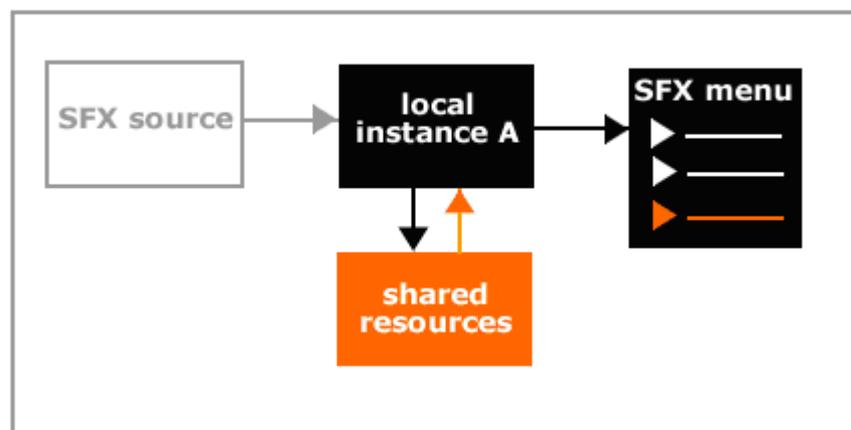


Figure 4: Local Instance Shared Resource

- **Scenario 2** – The central office maintains the SFX sources. For each of these sources, the shared resource instance has control over the design and layout of the SFX menu.

First, an SFX source sends an OpenURL to the shared resource instance. This instance uses the IP institute to determine which institution the user is coming from and which corresponding SFX instance should be used (instance A in image below). More information about the IP institute can be found in the *SFX Advanced User Guide*.

In the second step, SFX forwards the OpenURL to the SFX instance of this user (instance A) and receives information about services. After receiving this information, the shared resources instance determines which services both local and remote to present in the SFX menu screen.

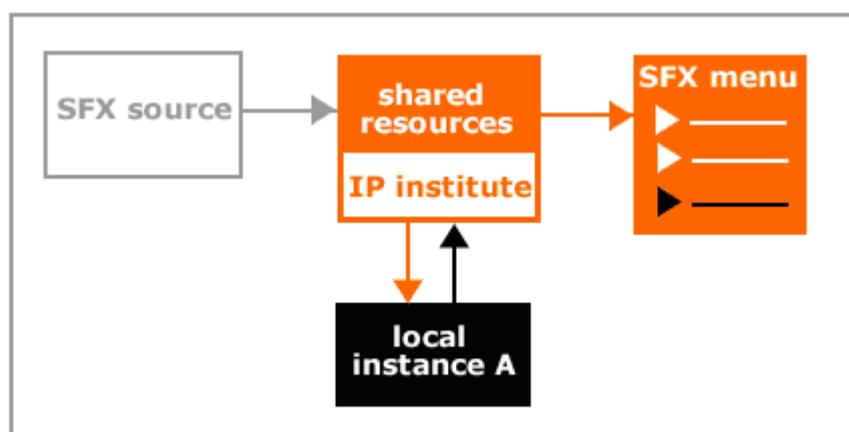


Figure 5: Central Office Maintains the SFX Source

NOTES:

- Both scenarios can exist in the same SFX installation. While some SFX sources can be controlled by the central office, others can be under the local control of each institution.
 - The different SFX instances do not need to exist on the same server, but can be located on different SFX servers.
 - It is possible to create A-Z lists of electronic journals which include both the local and shared resources.
-

Model 3: Single SFX Instance - Multiple Institutions Sharing the Same Instance

This model is characterized by the following:

- Multiple institutions in a consortium share the same SFX instance.
- The instance is centrally managed and contains the shared data as well as the non-shared data.
- Implementation of this model depends on a large number of the resources and services being shared by the different institutions (85-90%).

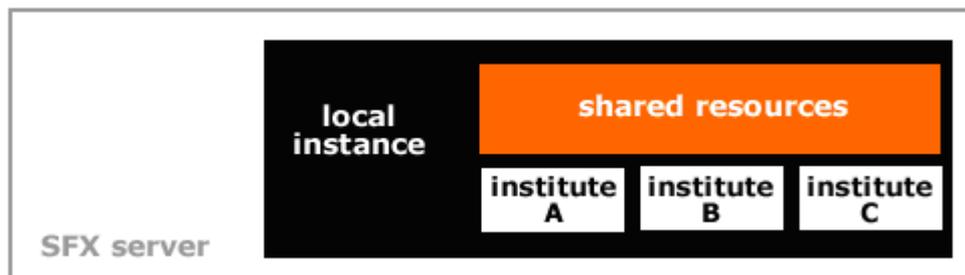


Figure 6: Single SFX Instance - Multiple Institutions Sharing the Same Instance

In this model, all the institutions sharing the local instance share:

- The design and layout of their SFX menu
- The description of services
- Logic rules

However, each institution can specify their own proxy server, username, and password information for specific resources. It is also possible to create institution specific A-Z lists of electronic journals.

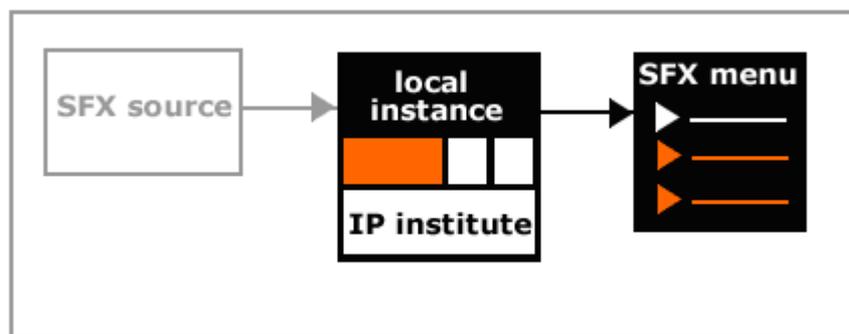


Figure 7: Institution Specific A-Z Lists

NOTE:

This model can be used in combination with model 1 and 2.

2

Setting Up Model 2: Using a Shared SFX Instance

This section includes:

- [Overview](#) on page 11
- [General Set Up](#) on page 12
- [Using the IP Institute Table to Handle Requests Sent to a Shared Instance](#) on page 18
- [Statistics in Model 2](#) on page 23
- [A-Z List and Institutes Using Consortium API](#) on page 24
- [Rapid Service Indicator Index and Consortia API](#) on page 24
- [Elsevier ScienceDirect Autoloader and Consortia API](#) on page 26

Overview

In a consortium environment, the SFX KnowledgeBase content can be shared between different institutions. In the following setup, shared services set up in a specific SFX instance can be added to the SFX menu screen of a different SFX instance regardless of whether or not it is on the same SFX server.

The different steps of this setup are shown in [Figure 8](#). First, an SFX source sends an OpenURL to an SFX instance. The instance forwards this OpenURL to a second SFX instance, called a shared instance, and receives information about available services in that instance. After receiving the information, the instance

determines which services, both local and shared, to present in the SFX menu screen.

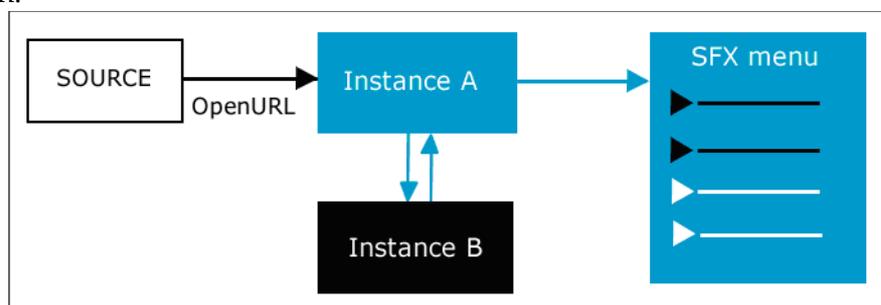


Figure 8: Including Services in the SFX Menu Screen from a Shared SFX Instance

The display logic rules that are applied to the SFX menu, the order of services, and the grouping of services into basic and advanced sections are determined by the instance that initially receives the OpenURL. This is the instance that has control over the display of services from both the local and shared instances.

However, the following display properties are controlled by the instance when the service is active: the target and target service public name, which target displayer is used, and the site up/down indication.

If the same service is active in both the local and shared instance, deduplication occurs and only one service is displayed in the SFX menu. The service from the instance that initially receives the OpenURL is displayed and the service from the instance to which the OpenURL request is forwarded is suppressed from the SFX menu.

General Set Up

The set-up for this feature requires you to define the relationship between the two instances. (These two instances can be on the same SFX server or on different servers):

- **Instance A** – Responsible for the layout of the SFX menu screen, the order of services, etc. This instance determines which services from the additional services instance to display in the SFX menu screen.
- **Instance B** – Some or all services of this instance are included in the SFX menu screen of instance A.

To set up this feature:

- 1 In instance A, go to the following configuration file:

```

/exlibris/sfx_ver/sfx4_1/<instance A>/config/
ctx_object.config:
  
```

- 2 By default, the `consortium` section is set to active `N`. Change this section to active `Y` and save the configuration file.

```
Section "consortium"  
    # Enable/disable fetching from consortium members  
    active      "Y"  
EndSection
```

- 3 Configure the relationship between instance A and B in the following file in instance A:

```
/exlibris/sfx_ver/sfx4_1/<instance A>/config/  
consortium.config
```

This configuration file in instance A contains information about instance B. Below is an example of a `consortium.config` file where instance A is called `arts` and instance B is called `central`. Configure the

`consortium.config` file of the local instance to refer to the shared instance as in the following example.

The configuration file in the `arts` instance looks like this:

```
# List of consortium members
Section "consortium_members"
central
EndSection

# List of consortium members assigned to each institute
# When this section is present, the
# 'consortium_members' section is skipped.
# COMMENT OUT THIS SECTION if not working with an
# institute environment,
# or if the consortium connections are not dependent on
# institutes
#Section "consortium_members_by_institute"
#   # format: <institute>  "<member>,[<member>]"
#   # _all_ lists the consortium members for those
#   request without institute
#EndSection

# SOAP namespace for each member
Section "member_namespace"
central    "SFXMenu/SOAP"
EndSection

# Gateway where SOAP services are exposed
Section "member_gateway"
central    http://10.1.2.225:3410/central/cgi/core/soap/
sfxmenu.cgi
EndSection
```

```
# SFX Resolver assigned to the services retrieved from
a member
Section "member_resolver_gateway"
central http://10.1.2.225:3410/central/cgi/core/soap/
sfxresolver.cgi
EndSection

Section "member_resolver_namespace"
central "SFXResolver/SOAP"
EndSection

# To avoid infinite loops:
# Do not try to retrieve services from <member>
# if the request comes from <IP> [/<member>]
Section "member_connection_constraint"
central "10.1.2.225:3410/central"
EndSection

# When connecting to <member>, identify me as <member>
Section "member_self_name"
central arts
EndSection

# When connecting to <member>,
# only retrieve <service>,<service>,...
Section "allowed_services"
central "getFullTxt,getHolding,getDocumentDelivery"
EndSection

# Before connecting to <member>,
# apply this threshold
# if threshold fails, don't connect
Section "threshold"
# central "$obj->need('rfr.id', 'eq', 'ALEPH')"
EndSection
```

```
#Do not store statistics in remote unless
store_in_remote is set to 1
Section "statistics_storage"
store_in_remote "0"
EndSection

#####
# Configuration and connection params for RSI #
#####

Section "rsi_consortium_members"
central
EndSection

Section "rsi_member_gateway"
central "http://10.1.2.225:3410/central/cgi/core/rsi/
rsi_consortia.cgi"
EndSection

Section "rsi_consortium_params"
    ready_wait          300
    max_waits           5
    fetch_timeout      60
EndSection

#####
# Configuration and connection params for A-Z eBooks /
iRSI #
#####
# schema used for replication from remote shared
(usually shared_<shared_instance_name>)
Section "schema_for_replication"
#     shared_remote_instance_name
EndSection

Section "shared_instance_name"
# When configuring a local instance forwarding requests
to a shared instance, provide shared instance name.
    central
EndSection
```

```

Section "shared_type"
# When configuring a local instance forwarding requests to the
shared instance:
# specify 'local' in case the shared instance exists in the
same SFX installation as the local instance
# specify 'remote' in case the shared instance exists in a
different installation than the local instance
    local
EndSection

Section "is_shared"
# When configuring a shared instance to which local instances
are forwarding requests, set value to 'Y'
#     N
EndSection

Section "remote_shared_of"
# This section should be configured only in a shared instance
where local instances exist on a different SFX server or
# SFX installation that the shared instance (this is called
'remote shared' set-up).
# Fill in the servername and solr port of each SFX
installation which contains local instances
#Example:
#     http://il-sfx08:8983
#     http://il-sfx02.exlibrisgroup.com:80

#     http://<server_name>:<solr_port>
EndSection

```

The following provides an explanation of each section of the `consortium.config` file:

- `consortium_members` - contains information about the name that is used for instance B (the instance to which you want to forward a request). If the instance name is meaningful, we recommend using the instance name as the consortia member name.

NOTE:

Commenting out (adding a # to the beginning of the line) a previously defined instance name turns off getting services for that instance.

- `member_namespace`, `member_gateway`, `member_resolver_gateway`, and `member_resolver_namespace` - specify the consortium member name defined in `consortium_members` and specify its server IP (or host name) and instance name.
- `member_connection_constraint` - used with the bi-directional API model. For more information, see [Appendix A: Set-Up for Different Model 2 Scenarios](#) on page 39.

- `member_self_name` – specify the name of instance A. This is the name that is communicated with instance B when requesting information about additional services.

Once instance B receives information about the instance requesting additional services, institute settings in instance B can restrict targets based on institute information. In order for this to work, `member_self_name` needs to be defined as an institute in instance B.

- `allowed_services` – define the types of services that are added from instance B, for example, both `getFullTxt` and `getHolding` services.
- `threshold` – set additional thresholds to the mechanism of getting services from instance B.
- `rsi_consortium_members`, `rsi_member_gateway` and `rsi_consortium_params` – used to build the Rapid Service Indicator index with shared resources included. More information about the Rapid Service Indicator index can be found in the **Rapid Service Indicator** section of the *SFX System Administration Guide*.

For a description of the parameters for the A-Z eJournals List and eBook Search indexes in the `consortium.config` file, see [Configuration for Consortia API - eBook Search](#) on page 44.

More information on creating the Rapid Service Indicator index can be found in [Rapid Service Indicator Index and Consortia API](#) on page 24.

For information on configuring the eBook Search and RSI indexes, see [Appendix B: eBook Search Set-Up for Consortium Model 2](#) on page 43.

Using the IP Institute Table to Handle Requests Sent to a Shared Instance

When SFX sends an OpenURL to a shared instance, the shared instance uses the IP address of the local instance located in the IP Institute table to send the response back to the local instance.

First, an SFX source sends an OpenURL to an SFX instance (instance A in [Figure 9](#)). This instance uses the IP institute to determine from which institute the user is coming and which corresponding SFX instance should be used

(instance B). More information about the IP institute can be found in the **Institute Feature** section of the *SFX Advanced User's Guide*.

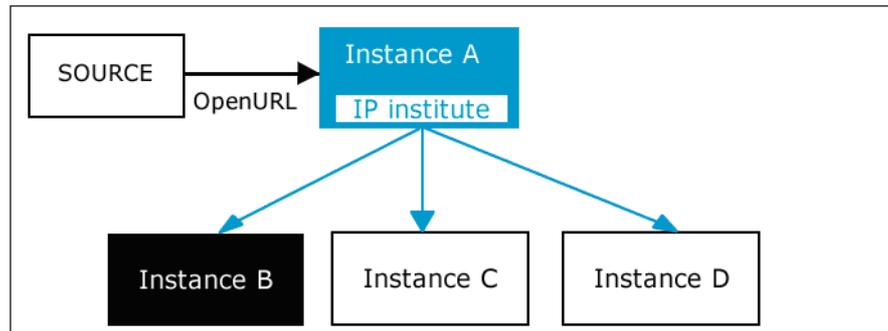


Figure 9: Using the IP Institute

In the second step, SFX forwards the OpenURL to this SFX instance (instance B) and receives information about available services. After receiving this information, instance A determines which services, both local and shared, to display in the SFX menu.

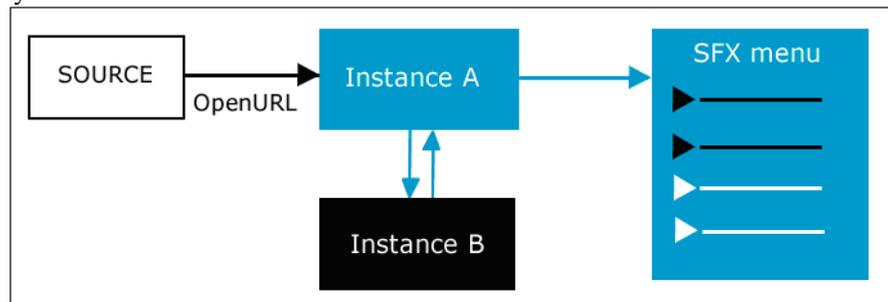


Figure 10: SFX Menu from Local and Remote Instances

To set up the IP Institute functionality:

- 1 Turn on this feature in instance A by changing the settings of the following configuration file:

```
/exlibris/sfx_ver/sfx4_1/<instance A>/config/  
ctx_object.config:
```

- 2 By default, the section consortium is set to active N. Change this section to active Y and save the configuration file.

```
Section "consortium"  
    # Enable/disable fetching from consortium members  
    active      "Y"  
EndSection
```

- 3 Define as institutes in instance A each of the instances to which the OpenURL needs to be forwarded (instance B, C, etc.). More information about defining institutes can be found in Section ?5 Institute Feature.

4 Configure the relationship between all the instances in the consortium.config file of instance A located at:

```
/exlibris/sfx_ver/sfx4_1/<instance A>/config/  
consortium.config
```

The following is an example of a consortium.config file in an institute environment where instance A is called central, instance B is called arts, and instance C is called sciences.

The following is an example of the configuration file of the central instance:

```
# List of consortium members  
#Section "consortium_members"  
#EndSection  
  
# List of consortium members assigned to each institute  
# When this section is present, the  
'consortium_members' section is skipped.  
# COMMENT OUT THIS SECTION if not working with an  
institute environment,  
# or if the consortium connections are not dependent on  
institutes  
Section "consortium_members_by_institute"  
# # format: <institute> "<member>,[<member>]"  
arts "arts"  
sciences "sciences"  
# # _all_ lists the consortium members for those  
request without institute  
EndSection  
  
# SOAP namespace for each member  
Section "member_namespace"  
arts "SFXMenu/SOAP"  
sciences "SFXMenu/SOAP"  
EndSection
```

```
# Gateway where SOAP services are exposed
Section "member_gateway"
arts      "http://10.1.2.225:3410/arts/cgi/core/soap/
sfxmenu.cgi"
sciences "http://10.1.2.225:3410/sciences/cgi/core/
soap/sfxmenu.cgi"
EndSection

# SFX Resolver assigned to the services retrieved from
a member
Section "member_resolver_gateway"
arts      "http://10.1.2.225:3410/arts/cgi/core/soap/
sfxresolver.cgi"
sciences http://10.1.2.225:3410/sciences/cgi/core/
soap/sfxresolver.cgi
EndSection

Section "member_resolver_namespace"
arts      "SFXResolver/SOAP"
sciences "SFXResolver/SOAP"
EndSection

# To avoid infinite loops:
# Do not try to retrieve services from <member>
#   if the request comes from <IP> [/<member>]
Section "member_connection_constraint"
#   arts  "10.1.2.225/arts"
#   sciences "10.1.2.235/sciences"
Endsection

# When connecting to <member>, identify me as <member>
Section "member_self_name"
arts  central
sciences central
EndSection
```

```
# When connecting to <member>,
# only retrieve <service>,<service>,..
Section "allowed_services"
arts "getFullTxt,getHolding,getDocumentDelivery"
sciences "getFullTxt,getHolding"
EndSection

# Before connecting to <member>,
# apply this threshold # if threshold fails, don't
connect
Section "threshold"
# arts "$obj->need('rfr.id', 'eq', 'ALEPH')"
# sciences "$obj->need('rfr.id', 'eq', 'ALEPH')"
EndSection

#####
# Configuration and connection params for RSI #
#####

Section "rsi_consortium_members"
arts
science
EndSection

Section "rsi_member_gateway"
arts "http://10.1.2.225:3410/arts/cgi/core/rsi/
rsi_consortia.cgi"
science "http://10.1.2.225:3410/science/cgi/core/rsi/
rsi_consortia.cgi"
EndSection

Section "rsi_consortium_params"
    ready_wait          300
    max_waits           5
    fetch_timeout       60
EndSection
```

- **consortium_members** – Do not use this section. for institute set-up, use the **consortium_members_by_institute** section instead.

- `consortium_members_by_institute` – List here each of the institutes defined in the `central` instance to which the OpenURL should be forwarded.

NOTE:

By commenting out (adding a # to the beginning of the line) a previously defined instance name, you turn off getting services from this instance.

- `member_namespace`, `member_gateway`, `member_resolver_gateway`, and `member_resolver_namespace` – specify the consortium member names defined in `consortium_members_by_institute`, and specify the server IP and instance name for each institute.
- `member_self_name` – specify the name of instance A. This is the name that is communicated with instance B and C when requesting information about additional services.

Once instance B receives information about the instance requesting additional services, institute settings in instance B can restrict targets based on institute information. In order for this to work, the `member_self_name` needs to be defined as an institute in instance B.

- `allowed_services` – define the types of services that are added from instance B, for example, when both `getFullTxt` and `getHolding` services from instance B are included in the SFX menu screen of instance A.
- `threshold` – set additional thresholds to the mechanism of getting services from another instance (instance B).
- `member_connection_constraint` – used with the bidirectional API model. For more information, see [Appendix A: Set-Up for Different Model 2 Scenarios](#) on page 39.
- `rsi_consortium_members`, `rsi_member_gateway` and `rsi_consortium_params` – used to build the Rapid Service Indicator index with shared resources included. More information about the Rapid Service Indicator index can be found in the **Rapid Service Indicator** section of the *SFX System Administrator Guide*.

More information on creating the Rapid Service Indicator index can be found in [Rapid Service Indicator Index and Consortia API](#) on page 24.

Statistics in Model 2

By default, requests forwarded from a local instance are not stored in the statistics tables of the shared instance, but only in the local SFX instance together with all the other usage data of that local instance.

It is possible to track requests coming through consortia API in the shared instance, which can be used to track requests coming from other instances.

To activate this option, change the following section in the `./config/consortium.config` file of the local instance:

```
Section "statistics_storage"  
    store_in_remote "1"  
EndSection
```

If the parameter `store_in_remote` is set to 1, requests and clickthroughs that were forwarded to the shared instance are recorded in the shared instance statistics tables.

To run statistics queries on usage in the shared instance, note that the institute names of each instance to which requests are forwarded have to exist in the shared instance and each `member_self_name` in the local `config/consortium.config` file has to correspond to the institute name in the shared instance.

NOTES:

- When this option is set to 1, all requests and clickthroughs are still stored in the local instance statistics tables, so the instance statistics data contains the complete data on usage.
 - If the `statistics_storage` section is not defined in the instance, the default setting, 0, is used. To set the value to 1, add the `statistics_storage` section shown above to the `config/consortium.config` file of the local instance.
-

A-Z List and Institutes Using Consortium API

For the A-Z list, no configuration is needed in `consortium.config`.

Since the RSI index is used to build the A-Z list, build the RSI index before creating the A-Z list.

Make sure that all of the institutes and groups for which you want to create A-Z lists that include services from the shared instance are defined in the shared instance (including the group assignments for each institute).

Rapid Service Indicator Index and Consortia API

The Rapid Service Indicator index is used by SFX and – via the RSI API – other systems to present FullText indicator information. More information about the Rapid Service Indicator index can be found in the **Rapid Service Indicator** section of the *SFX System Administrator Guide*.

To make sure that the Rapid Service Indicator index contains both local and shared resources:

- 1 Configure the following section in the `config/consortium.config` file of the local instance to build the Rapid Service Indicator index with shared resources included:

```
#####
# Configuration and connection params for RSI #
#####

Section "rsi_consor_tium_members"
#   remote_instance_name
EndSection

Section "rsi_member_gateway"
#   remote_instance_name      "http://
remote_host_name:3410/remote_instance_name/cgi/core/
rsi/rsi_consor_tia.cgi"
EndSection

Section "rsi_consor_tium_params"
    ready_wait          300
    max_waits           5
    fetch_timeout      60
EndSection
```

- 2 Use the Server Admin Utility to generate an RSI index in the shared instance. Start with the shared instance from which you are receiving shared resources.
- 3 Use the Server Admin Utility to generate an RSI index in the local instance. Based on the `consortia.config` file, SFX puts the data from the RSI index of the shared instance into the RSI index of the local instance.

NOTE:

The RSI index is used also when building the A-Z index.

- 4 Restart MySQL after the RSI index has been set up for the shared instance if a new instance is created in an existing SFX installation to be used as a shared instance.

For more information on stopping and then starting MySQL, see the **Start Stop Services** section of the *SFX System Administration Guide*.

Elsevier ScienceDirect Autoloader and Consortia API

All Elsevier Sciencedirect target activation is done in the local instance in order to be able to use the Elsevier autoloader. The Elsevier file contains all holdings per institution, regardless of whether the books/journals are purchased by the consortium or locally.

For more information on the Elsevier ScienceDirect autoloader, refer to the **ELSEVIER_SD/Elsevier SD** section of the *SFX Target Configuration Guide*.

3

Multiple Institutions Sharing One Instance

This section includes:

- [Overview on page 27](#)
- [IP Institute and Institute Thresholds on page 27](#)
- [Support for Multiple Proxies on page 28](#)
- [Storing Institute-Specific Linking Parameters for a Particular Target on page 28](#)
- [Creating HTML e-Journal Lists on page 31](#)
- [Configuring the Elsevier ScienceDirect Autoloader on page 32](#)

Overview

It is possible for multiple institutions or members of a consortium to use one SFX instance, if the electronic resources they have access to are very similar. This SFX instance contains information about electronic resources shared by the different institutions. A number of options in the SFX software need to be configured for this particular setup.

IP Institute and Institute Thresholds

The IP Institute table is used to store information about the institution or campus to which the user belongs. This is useful in consortia or group environments, where the context of the user affects access to different SFX services.

Currently, institute information about a particular user can be derived from:

- The user's IP address

- A cookie in the user's browser
- Information sent by another SFX instance

In all of these methods, the institute names are added to the SFX ContextObject. These names can then be used by SFX, when determining which targets should be shown for which institute.

For more information on the Institute Feature, see the **Institute Feature** section of the *SFX Advanced User's Guide*.

Support for Multiple Proxies

Each of the institutes using the same instance can specify their own proxy set-up. This is done from the **Proxy Settings** section, located at **Administrator Tools > Menu Configuration > Proxy/CrossRef** in the SFX Admin center.

This section allows the definition of:

- The default proxy – This proxy is used in non-consortia environments and when no institute specific proxy is defined.
- Institute specific proxies

Configure all proxies as described in the **Proxy/Crossref** section of the *SFX General User's Guide*.

Storing Institute-Specific Linking Parameters for a Particular Target

Institute-specific linking parameters are stored in a separate table in the SFX database.

This table consists of the following fields:

- The target service name
- The Parameter – The value of the parameter (in the given example: `$$EH_USER_ID`) matches the `$$` parameter in the `parse_param` field. When SFX creates a specific target URL, the `$$` parameter in the `parse_param` field is replaced with the corresponding Value.
- The Value – The actual user name, password, or other local parameters for the specific institute.

To set up institute-specific linking parameters:

- 1 From the KBManager section in the SFX Admin center, click **Targets**. The list of targets is displayed.

#	Target	Threshold	Modified	Active	Action
1	856_URL	s	2009/11/18	✓	c
2	AAA_HIGH_LCL	s P	2010/01/11	✓	c
3	AAA_HIGHER_LCL	s P	2010/01/11	✓	c
4	AA_TEST_LCL	s P	2009/12/11	✓	c
5	ABC_CLIO (2)				
6	ACCESS_MEDICINE_EBOOKS	s P	2009/11/18	✓	c
7	ACLS_HUMANITIES_EBOOK	s P	2009/11/18	✓	c
8	ACM_DIGITAL_LIBRARY (3)				
9	ACQ_EX_LIBRIS_ALEPH	s	2009/11/18	✓	c
10	ACTA_PRESS	s P	2009/11/18	✓	c

Figure 11: Target List

- 2 Click the Service button **s** next to the target service to which you want to add different linking parameters for each of the different institutes.
- 3 Click the L/P button **L/P**. The List of Linking Parameters dialog box is displayed.

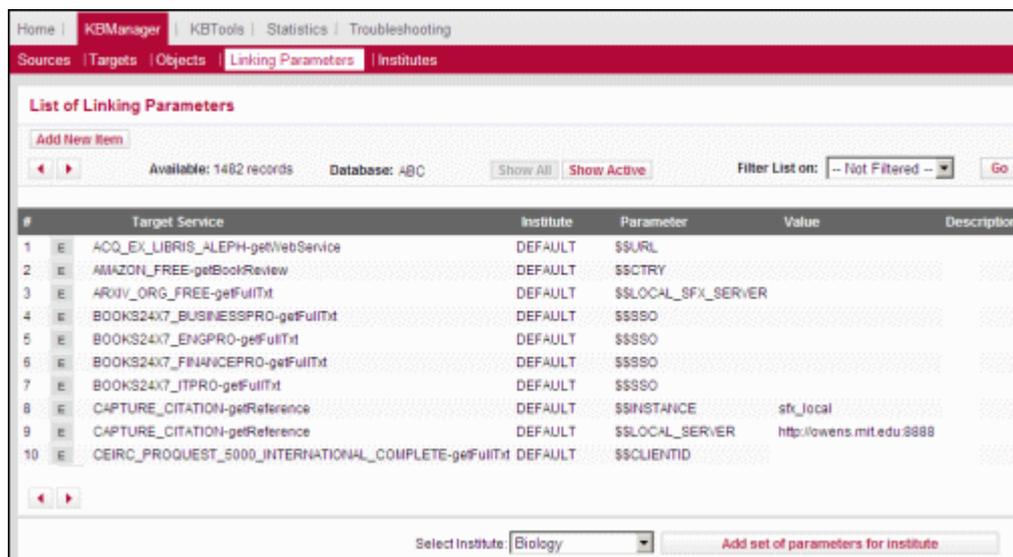


Figure 12: List of Linking Parameters

- 4 Select an institute from the drop-down list.
- 5 Click the **Add Set of Parameters for Institute** button. The Edit Linking Parameters dialog box is displayed:

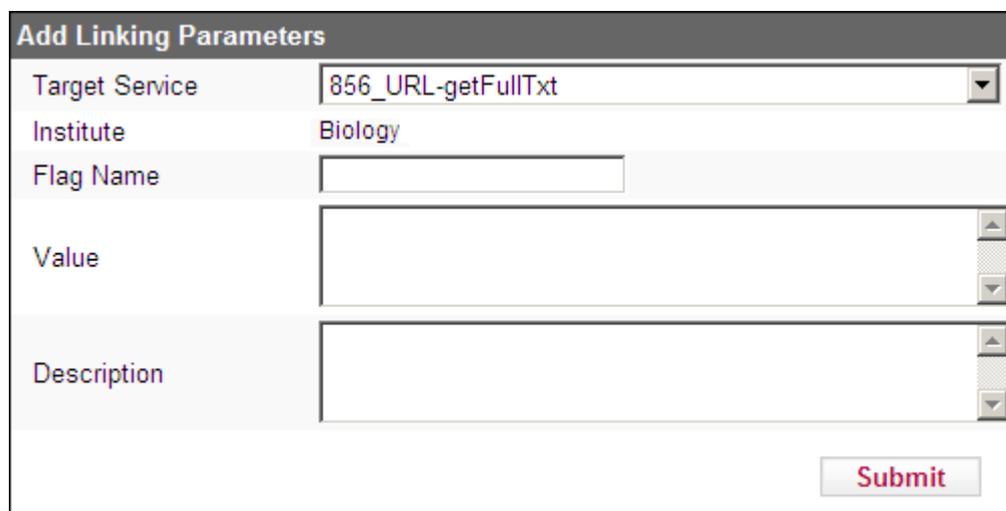


Figure 13: Add Linking Parameters

- 6 Fill in the Flag Name.
- 7 Fill in the authentication information in the **Value** field.
- 8 Click **Submit** to save the information.

Repeat this procedure for each of the institutes for which specific user name and password information or other local information needs to be specified.

Creating HTML e-Journal Lists

Some features, such as CJK support, are not supported in the HTML export format. It is recommended to use the searchable and browsable SFX A-Z list instead. For more information, refer to the **A-Z List** section of the *SFX General User's Guide*.

Creating Complete e-Journal Lists

To create a complete e-journal list, it is necessary to merge:

- The information about shared resources, stored in the remote instance
- The information about local resource, stored in the local instance

To create a complete e-journal list:

- 1 Create a tab-delimited e-journal list for shared resources.
 - a Log on to the SFXAdmin Center of the shared resource instance.
 - b From the **KB Tools** section, click **Export Tool>Advanced Export Queries**.
 - c Restrict the query to your instance.
 - d Select the **TXT** output format.
 - e Select an export type.
 - f Select export all active **getFullTxt** object portfolios
 - g Select Export from **ALL** targets.
 - h Save the TXT file on your local PC.
- 2 Create a complete e-journal list for shared and local resources:
 - a Log on to the SFXAdmin Center of the local resource instance.
 - b From the **KB Tools** section, click **Export Tool>Advanced Export Queries**.
 - c Select the **HTML** or **XML** output format.
 - d Select an object type.
 - e Specify an additional export file - the TXT list which was created in step 1.
 - f Select export all active **getFullTxt** object portfolios

g Select Export from **ALL** targets.

The HTML e-journal list that is created contains information from both SFX instances.

Creating Institute-Specific e-Journal Lists

It is possible to create an institute-specific e-journal list, which means the institute thresholds specified in the SFX instance are taken into account when creating a specific e-journal list.

To do this, type the name of the institute or institutes in the **Restrict to the following institutes/groups (optional)** box . The generated file contains a list of object portfolios that are active and available for those particular institutes. Institute thresholds can be set at the Target, Target Service, or Object Portfolio level in KBManager.

Configuring the Elsevier ScienceDirect Autoloader

Each institution receives a separate institutional token from Elsevier. The ServerAdmin Utility program can be configured to work with multiple holdings files per instance (one per institute), each with separate credentials.

For each institute, create a separate section where the institution token can be stored in the `config/elsevier_sd_autoloader.config_file`.

In the following example, the token for two institutes (Inst1 and Inst2) has been configured. A new section is created for each institute. The institute name is

appended to the section name, and the institute token is included in the parameter called `token`:

```
#####  
# Configuration per 'institute'  
# Use this section in case Elsevier provides a separate  
institutional_token per institute.  
# For each institute, create a new section, where institute  
name is appended to the section name.  
# 'active': indicates if the process will be run through the  
centralized job manager (0|1).  
# 'token': the token provided by Elsevier for the institute  
#####  
  
Section "institutional_token_Inst1"  
    active      0  
    token       "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"  
EndSection  
  
Section "institutional_token_Inst2"  
    active      0  
    token       "yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy"  
EndSection
```

For more information on the Elsevier ScienceDirect autoloader, refer to the **ELSEVIER_SD/Elsevier SD** section of the *SFX Target Configuration Guide*.

4

Locator.cgi

This section includes:

- **Overview** on page 35
- **General Set Up** on page 35
- **Locating the Appropriate SFX Instance Based on IP Range** on page 37
- **Locating the Appropriate SFX Instance Based on a Cookie** on page 37

Overview

The `locator.cgi` program on the SFX server is used when there is more than one local instance, but the SFX source (database vendor) is set up to only recognize one SFX base-URL. `locator.cgi` can then used to:

- Locate the appropriate SFX instance based on the user's IP range or cookie.
- Redirect the OpenURL that was sent to `locator.cgi` to the appropriate instance.

General Set Up

The `locator.cgi` program can be accessed from each instance using the following URL:

http://<sfx_server>:<port>/<sfx_instance>/cgi/core/locator.cgi

For example:

<http://demo.exlibrisgroup.com:3410/sfxlcl41/cgi/core/locator.cgi>

SFX sources store this URL instead of the base URLs of the different instances and send OpenURLs to this URL.

For example:

<http://demo.exlibrisgroup.com:3410/sfxlcl41/cgi/core/locator.cgi?sid=EBSCO:afh&genre=article&issn=0000-0019&date=2004&issue=12&volume=4>

Redirection to the correct instance on the same (or different) SFX server occurs based on the set-up in the following two configuration files:

- `locator.config_`
- `locator.dir`

Both configuration files can be found at:

`/exlibris/sfx_ver/sfx4_1/<sfx_instance>/config`

To set up `locator.cgi`, edit the `locator.config_` file. The `locate_by` section in the `locator.config_` configuration file is used to specify by which method (either IP range or cookie) the `locator.cgi` file determines the appropriate SFX instance. `locate_by` contains two parameters:

- IP
- Cookie

Type 1 next to the method you want to select.

In this example, the method chosen is IP:

```
Section locate_by
    IP          "1"
    cookie     "0"
EndSection
```

The `Default` section in `locator.config_` is used to specify the default SFX instance to be used if:

- The IP range of the user does not exist in `locator.dir`.
- The user does not have a cookie.

For example:

```
Section "Default"
    default_base_url    "http://
sfxserver.university.edu:3410/sfxlcl41"
EndSection
```

Locating the Appropriate SFX Instance Based on IP Range

In order to locate the appropriate SFX instance based on IP range, specify the IP range and the corresponding `BASE_URL` for the IP range in the `locator.dir` configuration file.

`locator.dir` is located at:

```
/exlibris/sfx_ver/sfx4_1/<sfx_instance>/config
```

The file contains the following columns:

- IP (range) – TCP/IP Network Address. This can be specified as a single address or a range of addresses. See the examples below.
- `base_url` – http://<sfx_server>:<port>/<sfx_instance>

For example:

#ip-range	#base-url
216.229.136.1	http://sfx1.exlibris-usa.com/demo
216.229.137.118-121	http://sfx1.exlibris-usa.com/test
216.229.138.*	http://sfx1.exlibris-usa.com/new

Based on this information, `locator.cgi`:

- Determines to which IP range the IP of the user belongs.
- Redirects the OpenURL to the appropriate `BASE_URL` that was specified for the IP range.

Locating the Appropriate SFX Instance Based on a Cookie

Request a cookie named `user-OpenURL` from the same domain as the `locator.cgi` script (that is, the same SFX server). The cookie contains a parameter named `BASE_URL` which determines the location of the appropriate SFX server and instance.

`locator.cgi` reads the content of the `user-OpenURL` cookie and using the information in the `BASE_URL` parameter, redirects the OpenURL sent from the source database to the appropriate SFX instance.

For example: A user who wants to direct OpenURLs to a particular instance sets a cookie such as:

http://sfx.lib.university.edu:3410/sfxlcl41/cgi/public/p3p_pushcookie.cgi?BASE-URL=http%3A//demo.exlibrisgroup.com%3A3410/demo&Redirect=http%3A//libraryhomepage.university.edu/databaselist.html

After setting the cookie, you can enter a source database that sends its OpenURLs to the `locator.cgi` script.

For example:

<http://demo.exlibrisgroup.com:3410/sfxlcl41/cgi/core/locator.cgi?sid=EBSCO:afh&genre=article&issn=0000-0019&date=2004&issue=12&volume=4>

is now be redirected (according to cookie information) to:

<http://demo.exlibrisgroup.com:3410/demo?sid=EBSCO:afh&genre=article&issn=0000-0019&date=2004&issue=12&volume=4>

A

Set-Up for Different Model 2 Scenarios

This section includes:

- **Scenario 1 – Requests Begin at Local Instance** on page 39
- **Scenario 2 – Bi-Directional API** on page 40
- **Scenario 3 – Chain of Remote Instances** on page 41
- **Scenario 4 – Consecutive Remote Instances** on page 42
- **Scenario 5 – Instance Specific Request Forwarding** on page 42

Note that for SFX Version 3, these models are provided as-is. It is recommended that the customer consult with their local Ex Libris support office if these are considered for local use.

Scenario 1 – Requests Begin at Local Instance

The OpenURL request is sent to a local instance (instance A) which then forwards the request to a shared instance (instance B). Instance A creates the SFX menu based on the information of the two instances.

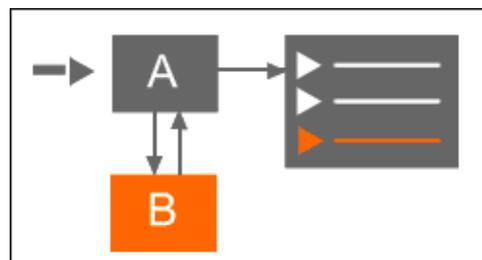


Figure 14: Requests Begin at Local Instance

In instance A:

- 1 Turn on the consortium setting in `ctx_object.config`.
- 2 Configure `consortium.config`.

In instance B - No configuration required.

Scenario 2 – Bi-Directional API

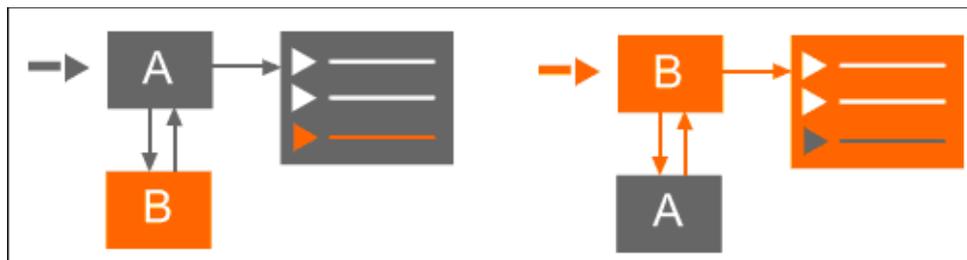


Figure 15: Bi-Directional API

In instance A:

- 1 Turn on the consortium setting in `ctx_object.config`.
- 2 Configure `consortium.config`.

In instance B:

- 1 Turn on the consortium setting in `ctx_object.config`.
- 2 Configure `consortium.config`.

Additionally, in each `consortium.config`, edit the infinite loop configurations.

Infinite loop configuration in instance B - Information that specifies that if a request is received from instance A, then the request should not be forwarded to instance A.

Infinite loop configuration in instance A - Information that specifies that if a request is received from B, then the request should not be forwarded to instance B.

Provide the name of the consortium member and the IP/instance name of the consortium member (but NOT the Apache port information). If necessary, provide both the external and the internal IP of the server. If the instance where configuration is performed receives open URLs from more than one local instance, list each local instance in the `member_connection_constraint` section.

Example:

```
Section "member_connection_constraint"  
    dv_3 "10.1.2.225/dv_3,10.1.2.231/dv_3,127.0.0.1/  
    dv_3"  
EndSection
```

Scenario 3 – Chain of Remote Instances

In this scenario, the request is forwarded from A to B and from B to C.

For the instance A, the second forward is hidden. In other words, if services are included from instance C they are not visible for instance A. It seems like the services come from instance B.

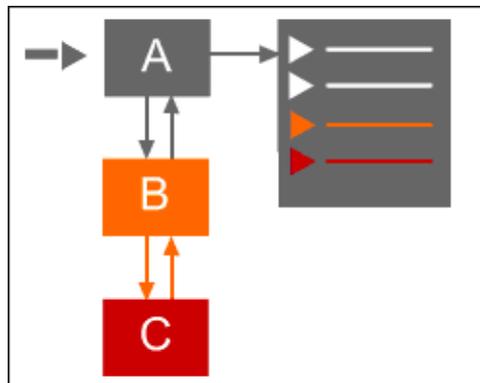


Figure 16: Chain of Remote Instances

In instance A:

- 1 Turn on the consortium setting in `ctx_object.config`.
- 2 Configure `consortium.config`.

In instance B:

- 1 Turn on consortium setting in `ctx_object.config`
- 2 Configure `consortium.config`.

In instance C – no configuration is required.

Scenario 4 – Consecutive Remote Instances

The requests are sent consecutively to B and C (not in parallel). So this slows down generation of the SFX menu screen for A.

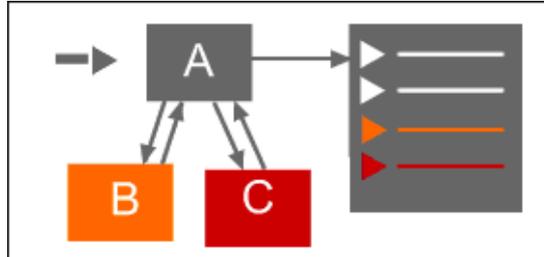


Figure 17: Consecutive Remote Instances

In instance A:

- 1 Turn on the consortium setting in `ctx_object.config`
- 2 Configure `consortium.config` for B and C.

In instance B and instance C – no configuration is required.

Scenario 5 – Instance Specific Request Forwarding

The requests are sent either to instance B or instance C, depending on to which institution the user belongs. The instance is determined based on IP or cookie.

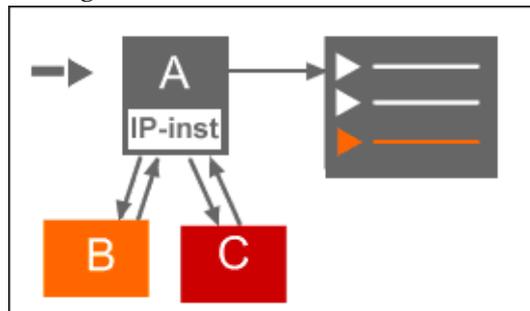


Figure 18: Instance Specific Request Forwarding

In instance A:

- 1 Turn on consortium setting in `ctx_object.config`
- 2 Configure `consortium.config` for B and C.

In instance B and instance C – no configuration is required.

B

eBook Search Set-Up for Consortium Model 2

This appendix describes how to set up the eBook Search feature for consortium model 2 - multiple SFX instances - including one instance for shared resources.

This section includes:

- [Overview](#) on page 43
- [Configuration for Consortia API - eBook Search](#) on page 44
- [Setting Up Replication for RSI in Both the Shared \(Master\) and Local \(Slave\) Instances](#) on page 47
- [Rebuilding A-Z eBooks in the Remote Shared Instance](#) on page 48
- [Troubleshooting and Debugging](#) on page 49

Overview

In Consortium model 2, each local institution has its own local instance and is responsible for its non-shared data. A central office manages the shared resource instance (shared instance) and is responsible for all shared data.

Interaction between the SFX instances results in the generation of the SFX menu and the A-Z List.

After completing the general eBook Search set-up (as described in **Appendix D eBook Search Set-Up Instructions** of the *SFX System Administrator Guide*), perform the following additional steps in the SFX environment if model 2 is set-up:

- Configure the shared and local instances configuration files – `config/ctx_object.config` and `config/consortium.config`. See [Configuration for Consortia API - eBook Search](#) on page 44.

If the shared instance does not exist in the same SFX installation as the local instance (this set-up is called a remote shared set-up), do the following, as well:

- Set-up replication for RSI. See [Setting Up Replication for RSI in Both the Shared \(Master\) and Local \(Slave\) Instances](#) on page 47.
- Rebuild the eBook search index (with a complete build) in the SFX installation of the remote shared instance. See [Rebuilding A-Z eBooks in the Remote Shared Instance](#) on page 48.

Configuration for Consortia API - eBook Search

Perform the configuration both in the local instance and the shared instance. The configurations are different depending on whether local and shared instances exist in same installation or not.

Configuration in the Local Instance Where Local and Shared Instances Exist in the Same SFX Installation

- 1 In the `config/ctx_object.config` file, set the `active` parameter of the `consortium` section to `Y`:

```
Section "consortium"
    active      "Y"
    <include:$ENV{SFXCTRL_HOME}/config/consortium.config>
    consortium_name      "my_consortium"
EndSection
```

NOTE:

This should be already set up, since it is required for SFX menu forwarding of requests to the shared instance.

- 2 Perform the following changes in the `config/consortium.config` file:
 - Set the `shared_instance_name` section to the name of shared instance that exists in the same installation:

```
Section "shared_instance_name"
    <instance>
EndSection
```

NOTE:

Use the instance UNIX name (not the instance Apache alias name).

- Set the `shared_type` section to `local`:

```
Section "shared_type"
    local
EndSection
```

- Leave the following sections blank:
 - `schema_for_replication`
 - `is_shared`
 - `remote_shared_of`

Configuration in a Shared Instance Where Local and Shared Instances Exist in the Same SFX Installation

- In the `config/consortium.config` file, set the `is_shared` section to `Y`:

```
Section "is_shared"  
    Y  
EndSection
```

- No configuration is required in the `config/ctx_object.config` file.

Configuration in the Local Instance Where Local and Shared Instances Do Not Exist in the Same SFX Installation

- 1 In the `config/ctx_object.config` file, set the `active` parameter in the `consortium` section to `Y`:

```
Section "consortium"  
    active      "Y"  
    <include:$ENV{SFXCTRL_HOME}/config/consortium.config>  
    consortium_name    "my_consorrtium"  
EndSection
```

NOTE:

This should be already set up, since it is required for SFX menu forwarding of requests to the shared instance.

- 2 In the `config/consortium.config` file:

- The `schema_for_replication` section looks like the following:

```
Section "schema_for_replication"  
    <shared_remote_instance_name>  
EndSection
```

Replace `<shared_remote_instance_name>` with `shared_<name of shared instance>`.

- Set the `shared_instance_name` section with the name of the shared instance:

```
Section "shared_instance_name"  
    <instance>  
EndSection
```

NOTE:

Use the instance UNIX name (not the instance Apache alias name).

- Set the `shared_type` section with the value `remote`:

```
Section "shared_type"  
    remote  
EndSection
```

- Leave the following sections blank:

- `is_shared`
- `remote_shared_of`

Configuration in a Shared Instance Where Local and Shared Instances Do Not Exist in the Same SFX Installation

In the `config/consortium.config` file:

- set the `is_shared` section to `Y`:

```
Section "is_shared"  
    Y  
EndSection
```

- Set the `remote_shared_of` section with at least one value. Fill in the server name and solr port of each SFX installation that contains local instances. The default solr port is 8983:

```
Section "remote_shared_of"  
    http://il-sfx08:8983  
EndSection
```

- Leave the following sections blank:

- `schema_for_replication`
- `shared_instance_name`
- `shared_type`

Setting Up Replication for RSI in Both the Shared (Master) and Local (Slave) Instances

NOTE:

This set-up is only required if the local and shared instances do not exist in the same SFX installation. In this case, replication needs to be set up for the shared instance RSI date to ensure that the shared instance data is available to the local instances.

The following tables are replicated from the remote shared installation:

- KB_OBJECT_IDENTIFIERS
- RSI_MONOGRAPH_<shared_name>
- RSI_MONOGRAPH_INSTITUTION

- 1 For each instance, choose a unique `server_id` (one for the master instance and one for each slave SFX installation)
-

NOTE:

It is important to make sure that each `server-id` is unique. The `server-id` is stored in the `my.cnf` file of each SFX installation that is part of the replication - so if instances are added at a later stage, it is important to check that the new `server-id` does not yet exist.

- 2 On the master (remote shared instance machine)
 - a As a prerequisite, build the RSI for eBooks in the shared instance.
 - b As as the `sfxglb41` user, add the following to the `my.cnf` file in the `/exlibris/sfx_ver/sfx4_1/data` directory of the `sfxglb41` instance:
 - `server-id=<number1 - positive integer, should differ from slave(s)>`
 - `log-bin=/exlibris/sfx_ver/sfx4_1/data/log-bin`
 - `expire-logs-days=5`
 - `binlog-do-db=sfxglb41`
 - `binlog_format=ROW`
 - `log_bin_trust_function_creators=1`
 - c Restart MySQL.
 - d Run the setup script for the master.
 - e Enter `cd /exlibris/sfx_ver/sfx4_1/sfxglb41/admin/database/`
 - f Enter `perl replication_setup_master.pl`
 - g Follow the instructions given by the script.

- 3 On the slaves (all local/slave machines) as the `sfxglb41` user:
 - a Put the tar file with the master data created in step 2.d in the slave `/data` directory.
 - b Copy the `shared_data.tgz` file from remote machine to the local shared instance schema directory (`/exlibris/sfx_ver/sfx4_1/data/`).
 - c Run the setup script for the slave.
 - Enter `cd /exlibris/sfx_ver/sfx4_1/sfxglb41/admin/database/`
 - Enter `perl replication_setup_slave.pl`
 - Follow the instructions given by the script.
 - d Edit `my.cnf`:
 - `server-id=<number2 - positive integer, differ from master's and other slaves'>`
 - `replicate-rewrite-db=<global instance name on shared>'shared_<shared instance name>`
 - `replicate-wild-do-table= shared_<shared instance name>.IRSI_<shared instance name>%`
 - `replicate- do-table= shared_<shared instance name>.IRSI_INSTITUTIONS`
 - `replicate- do-table= shared_<shared instance name>.KB_OBJECT_IDENTIFIERS`
 - e In the `my.cnf` file, after `# Bin Log`, add the following:

```
log_bin_trust_function_creators=1
```
 - f Restart MySQL.

Rebuilding A-Z eBooks in the Remote Shared Instance

Build RSI eBooks and A-Z ebooks using the Server Admin Utility:

- 1 Select **7 AZ Index**.
- 2 Select **2 eBook Search index tools**.
- 3 Select **1 (Re)build the eBook Search index**.

Troubleshooting and Debugging

- To remove the lock file if the RSI building process has to be stopped:
 - a In the local instance, connect to MySQL as the `root` user and enter:

```
use <local instance db schema>
```

- b Select `control_value` from `CONTROL` where `control_group = 'rsi_monograph'` and `control_name='currently_running'`

```
+-----+
| control_value |
+-----+
| 1             |
+-----+
```

- c Update `CONTROL`. Set `control_value = 0` where `control_group = 'rsi_monograph'` and `control_name='currently_running'`

- d Check by selecting `control_value` from `CONTROL` where `control_group = 'rsi_monograph'` and `control_name='currently_running'`.

```
+-----+
| control_value |
+-----+
| 0             |
+-----+
```

- To recover from an error in replication on the slave instance:
 - a Log on as the master and connect to MySQL as the `root` user.
 - b Enter `show master status;`

```
+-----+-----+-----+-----+
| File          | Position    | Binlog_Do_DB    |
Binlog_Ignore_DB |
+-----+-----+-----+-----+
| log-bin.000004 | 1029359794 | sfxglb41,sfxglb41 |
+-----+-----+-----+-----+
```

- c Use the `file` and `position` information in the slave.
- d Go to the slave instance (`sfx1c141`) and log on to MySQL as the `root` user.
- e Enter `stop slave;`

f Enter the following:

```
CHANGE MASTER TO MASTER_HOST='il-sfx04',  
MASTER_USER='repl_slave', MASTER_PASSWORD='repl_slave',  
MASTER_LOG_FILE='<fill in 'show master status' File>',  
MASTER_LOG_POS=<Fill in 'show master status' Position>,  
master_port=3310;
```

g Enter `start slave;`

- In case of problems with activations not being shown in the local instance after rebuilding RSI in the shared instance:

a In the master and slave instances, log on to MySQL as the `root` user:

```
mysql -p<mysql_root_pwd> -uroot
```

b Enter the following:

```
'show master status' on master  
'show slave status' on slave
```

c Check that the position value is the same for both the master and slave instances.