

# MORPHEUS

BXF SERVICE CONFIGURATION

## Engineering Manual Supplement

2018-09-10

[www.grassvalley.com](http://www.grassvalley.com)

# Copyright and Trademark Notice

Copyright © Grass Valley Canada. All rights reserved.

Belden, Belden Sending All The Right Signals, and the Belden logo are trademarks or registered trademarks of Belden Inc. or its affiliated companies in the United States and other jurisdictions. Grass Valley, are trademarks or registered trademarks of Grass Valley Canada. Belden Inc., Grass Valley Canada, and other parties may also have trademark rights in other terms used herein.

# Terms and Conditions

Please read the following terms and conditions carefully. By using documentation, you agree to the following terms and conditions.

Grass Valley hereby grants permission and license to owners of to use their product manuals for their own internal business use. Manuals for Grass Valley products may not be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose unless specifically authorized in writing by Grass Valley.

A Grass Valley manual may have been revised to reflect changes made to the product during its manufacturing life. Thus, different versions of a manual may exist for any given product. Care should be taken to ensure that one obtains the proper manual version for a specific product serial number.

Information in this document is subject to change without notice and does not represent a commitment on the part of Grass Valley.

Warranty information is available from the Legal Terms and Conditions section of Grass Valley’s website ([www.grassvalley.com](http://www.grassvalley.com)).

Title	Morpheus BXF Service Configuration
Part Number	N/A
Revision	2018-09-10, 09:16

# toc

## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
<b>2</b>	<b>BXF Service Configuration.....</b>	<b>3</b>
	Terminology .....	3
	MAPP Terminology.....	3
	BXF Terminology.....	4
	Morpheus Terminology .....	4
	BXF Database Connection Strings.....	5
	Creating a BXF Service.....	5
	Configuring or Editing a BXF Service .....	7
	The BXF Message.....	8
	BXF IDs .....	8
	Translations.....	11
	MAPP Process IDs (PIDs) and PID Chains.....	12
	Allocating Process IDs (PIDs).....	12
	BXF Dates and Times .....	13
	BXF Schema Versions.....	13
	BXF Host Shell Services .....	14
	BXF Ardome SOAP Sender Service .....	15
	BXF AsRun Notifier Service.....	15
	BXF AsRun Notifier Service: Principle of Operation .....	16
	BXF Tab .....	17
	AsRun Tab.....	20
	Translation Tab .....	23
	Other Translations Tab .....	27
	Extras Tab .....	28
	BXF Content Transfer To Record Service.....	30
	BXF Content Transfer to Record Service: Principle of Operation.....	31
	BXF Content Transfer To Record Service Configuration Parameters .....	32
	BXF Database Maintenance Service .....	35
	BXF Database Maintenance Service: Principle of Operation.....	35
	BXF Database Maintenance Service Configuration Parameters .....	36
	BXF File Reader Service .....	37
	BXF File Reader Service: Principle of Operation.....	37
	BXF File Reader Service Configuration Parameters .....	38
	BXF File Writer Service.....	42
	BXF File Writer Service: Principle of Operation .....	42
	General Settings Tab.....	43
	Server Mappings Tab .....	46
	BXF Import Content Service .....	48
	BXF Import Content Service: Principle of Operation .....	48
	BXF Import Content Service Configuration Parameters .....	49

BXF Lock Service .....	53
BXF Lock Service: Principle of Operation .....	54
Configuring the BXF Lock Service .....	55
BXF Media Change Notifier Service .....	61
BXF Media Change Notifier Service: Principle of Operation .....	64
BXF Media Change Notifier Service General Configuration Parameters .....	65
BXF Media Change Notifier Service Filters .....	68
BXF Media Metadata Import Service .....	72
BXF Multiplex Message Service .....	73
BXF Multiplex Message Service: Principle of Operation .....	74
BXF Multiplex Message Service Configuration Parameters .....	75
BXF Query Service .....	77
BXF Query Service: Principle of Operation .....	78
Common Parameters .....	86
Schedule / AsRun: Translation Tab .....	89
Schedule / AsRun: Extras Tab .....	94
Schedule / AsRun: AsRun Tab .....	96
Schedule / AsRun: Other Translations Tab .....	99
Content (Item / Instance) Tab .....	101
BXF Schedule Change Notifier Service .....	103
BXF Schedule Change Notifier Service: Principle of Operation .....	103
Common Parameters .....	104
Translation Tab .....	107
Extras Tab .....	111
Destination Tab .....	114
Filters Tab .....	116
Other Translations Tab .....	128
BXF Schedule Service .....	129
BXF Schedule Service: Principle of Operation .....	130
BXF Schedule Configuration - Common Parameters .....	131
Channels Event Type Configuration Tab .....	136
Messages Special Actions Tab .....	143
Offline Editor Mode & Export.sch Tab .....	149
Other Translation Configuration Tab .....	152
Warning Report Configuration Tab .....	155
Synchronisation Tab .....	157
BXF SOAP Reader Service .....	160
BXF SOAP Writer Service .....	162
General Settings Tab .....	163
Server Mappings Tab .....	164
BXF Socket Reader Service .....	165
BXF Socket Reader Service: Principle of Operation .....	165
BXF Socket Reader Service Configuration .....	166
BXF Socket Writer Service .....	168
BXF Socket Writer Service: Principle of Operation .....	168
General Settings Tab .....	169
Server Mappings Tab .....	172
MAPP BXF Transfer Request Service .....	173
MAPP BXF Transfer Request Service: Principle of Operation .....	175
Content Transfer Update Message .....	177
General Settings Tab .....	178
Transfer Settings Tab .....	183
Destination By Up To Time Required Tab .....	190
Owner and Target Device Rules Tab .....	192
Extra Information at BXF Private Section Tab .....	195
Automatic Detect Completed Requests Tab .....	196
Filter Tab .....	199
Special Mode Tab .....	201
Locks Tab .....	204

# 1

## Introduction

The purpose of the Engineering Manual Supplement is to describe in full the details of new product features ahead of programmed releases of the Morpheus Engineering Manual.

It ensures that technical staff remain fully up to date with ongoing software and hardware developments, understand the purpose of the technology, and are able to implement new features as soon they become available.

The contents of this document will be published in full in the next release of the Morpheus Engineering Manual.



# BXF Service Configuration

The **B**roadcast **E**xchange **F**ormat (BXF) defines the format and content of XML messages for the exchange of data between broadcast systems.

## Terminology

An explanation of terms commonly used throughout this document.

### MAPP Terminology

**Asset Mail:** an application that responds to a Transfer Request message in order to facilitate the transfer of material between MAPP systems.

**Transfer Request (TR):** a MAPP message that is used in order to call for content to be moved or copied from one location to another.

**Record:** a set of fields in the database table that describe a single entity.

**Field:** one characteristic of a record - it may contain a value.

**Value:** a unique quality in a field that provides a description of a part of the record.

## BXF Terminology

**Content Transfer Request (CTR):** a message type that encapsulates a TR for delivery over a BXF environment.

**Element:** the building block of XML, containing descriptive information. The example below contains the following three elements: <StartDateTime>, <SmpteDateTime>, and <SmpteTimeCode>

```
<StartDateTime>
  <SmpteDateTime broadcastDate="2006-06-21">
    <SmpteTimeCode>15:42:03:20</SmpteTimeCode>
  </SmpteDateTime>
</StartDateTime>
```

**Attribute:** additional information that describes an element. In the example below, broadcastDate is an attribute of the <SmpteDateTime> element.

```
<StartDateTime>
  <SmpteDateTime broadcastDate="2006-06-21">
    <SmpteTimeCode>15:42:03:20</SmpteTimeCode>
  </SmpteDateTime>
</StartDateTime>
```

**Value:** the quality of an element or an attribute. In the example below, "2006-06-21" is the value for the broadcastDate attribute, and 15:42:03:30 is the value for the <SmpteTimeCode> element.

```
<StartDateTime>
  <SmpteDateTime broadcastDate="2006-06-21">
    <SmpteTimeCode>15:42:03:20</SmpteTimeCode>
  </SmpteDateTime>
</StartDateTime>
```

## Morpheus Terminology

**X & Y System:** dual redundant Morpheus environments.



## BXF Database Connection Strings

The BXF database connection string entered into the 'Deploy' tool is for the initial configuration; it does not provide database connectivity for the BXF agents. Each agent must be configured on an individual basis with its own connection string.

For information regarding the 'Deploy' tool, refer to the Morpheus Engineer's Manual.

For information related to the BXF agents, refer to their individual sections in this chapter.


---

Note: BXF agents communicate with Morpheus applications using the Microsoft .NET framework protocols.

---

## Creating a BXF Service

A BXF Service is created through the Shell Services Host, as follows:

1. In the 'Morpheus Shortcuts' folder, click on the 'Shell Services Host' shortcut 
2. In the 'Shell Services Host' window, click on **Service Settings** (Figure 2-1).

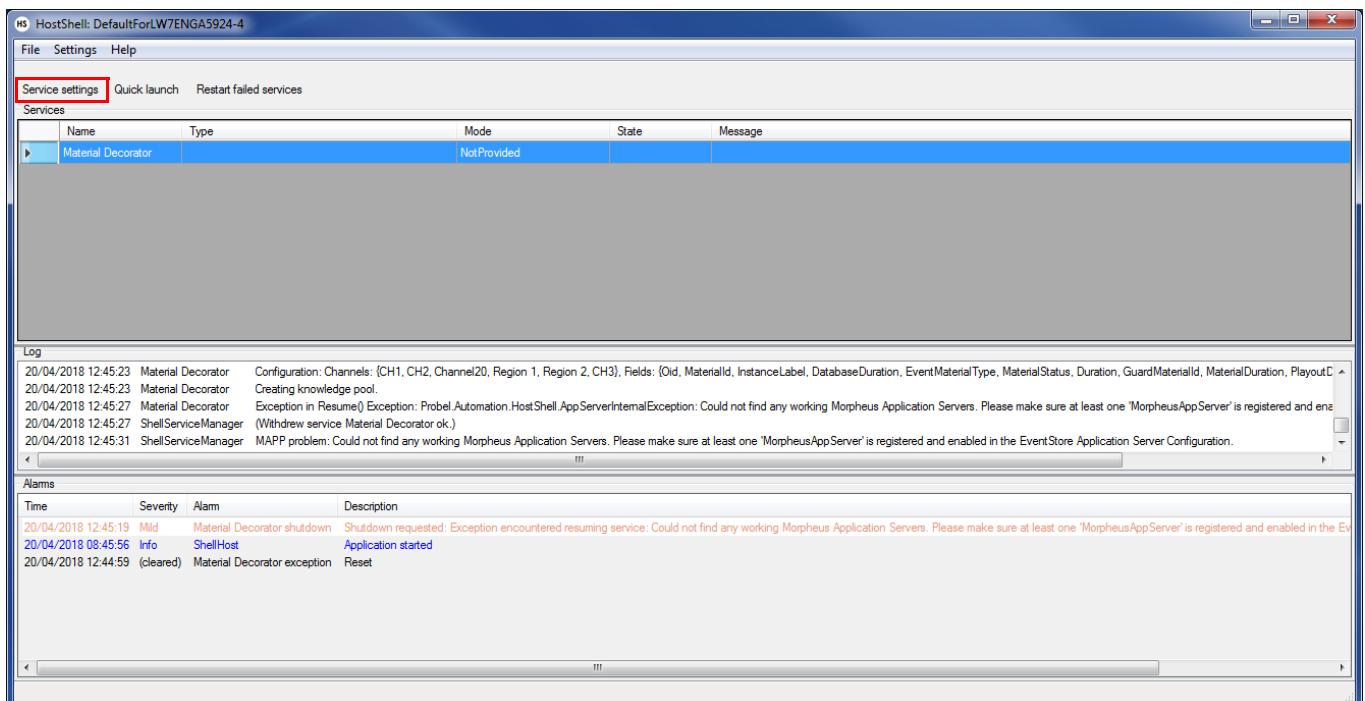


Fig. 2-1: Shell Services Host

The 'ShellHost Configuration' window is displayed (Figure 2-2).

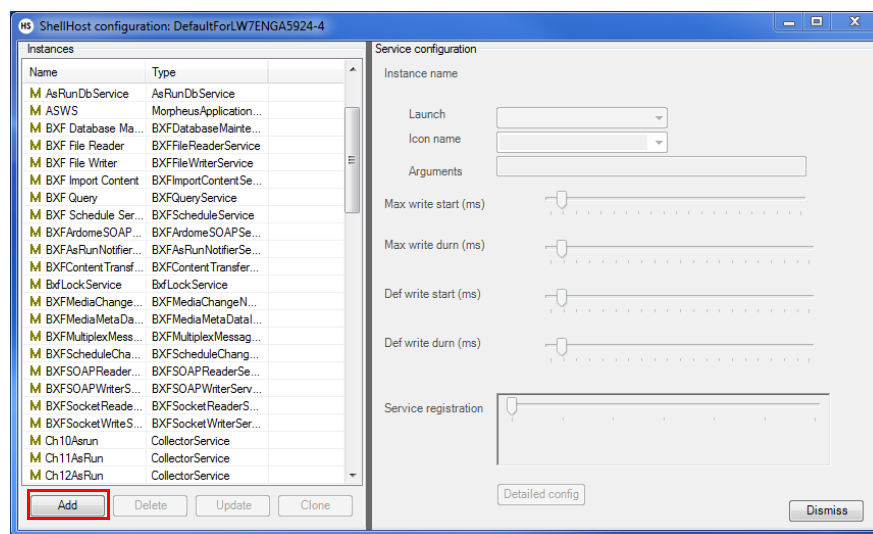


Fig. 2-2: ShellHost Configuration Window

3. Click on the **Add** button.

The 'Add Service...' window is displayed (Figure 2-3).

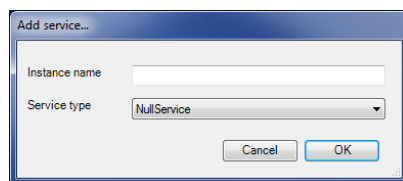


Fig. 2-3: Add Service...

4. In the 'Instance Name' text box, enter a unique name for the instance of the service that will be created.
  - a) From the 'Service Type' drop-down menu, select the type of service to create..
  - b) Click on **OK**.

The new service is added to the list of instances in the ShellHost Configuration window.

## Configuring or Editing a BXF Service

1. In the 'ShellHost Configuration' window, perform one of the following:
  - Double-click on the service in the list of instances.or
  - Left click on the service in the list of instances then click on the **Detailed Config** button
2. The service's configuration window will be displayed.

## The BXF Message

Supporting information relating to this implementation of the BXF standard.

### BXF IDs

The Bxflid is a Globally Unique Identifier (GUID) that uniquely labels an event within a BXF message for the purpose of performing 'Add', 'Update', and 'Remove' actions. If a Morpheus environment uses a BXF transport system then any event type can be configured with a Bxflid, although it is common to be configured only for top level events. A Bxflid parameter can be configured for a single event type, or for all event types.

Configure a Bxflid parameter for a single event type as follows:

1. Open the Configurator
2. Click on the **Configure Events** button.
3. Select the event type to configure.
4. Click on the 'All Parameter Definitions' tab.
5. Click on the **Add Parameter** button. The 'Modify Parameter Properties for a Single Event Type' window is displayed (Figure 2-4).

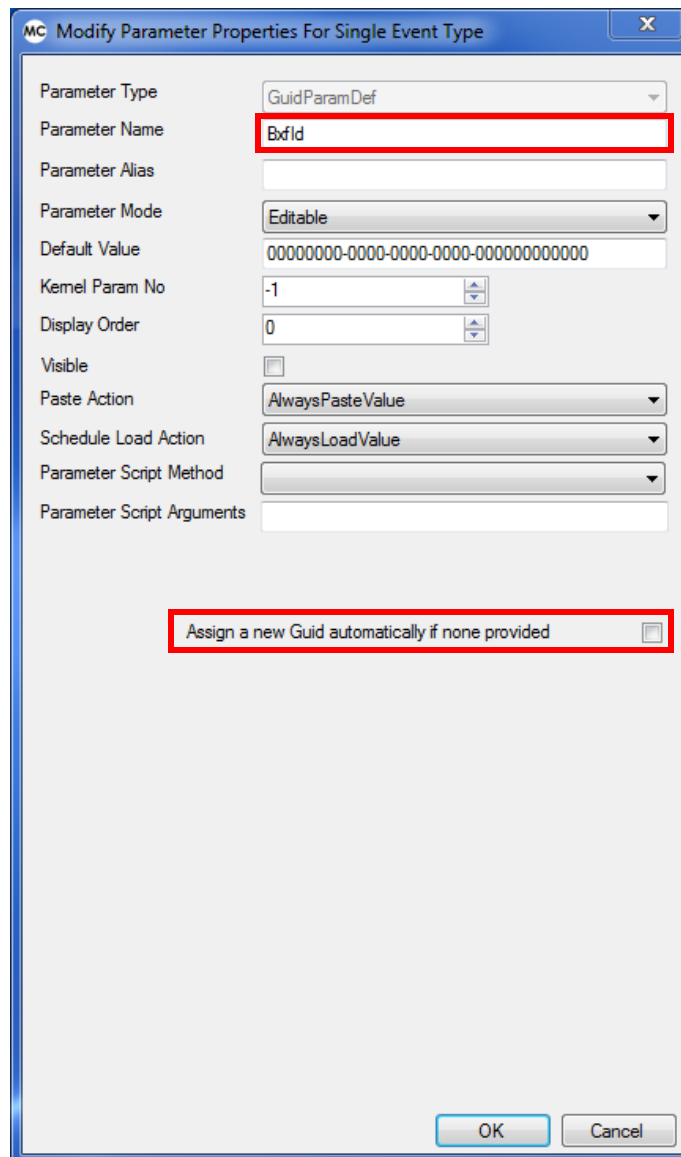


Fig. 2-4: Configuring a Single Event with a Bxfld

6. Select the 'Parameter Type' **GuidParamDef**.
7. Enter the 'Parameter Name' **Bxfld**.

---

Note: The spelling and case of *Bxfld* must be exactly as shown.

---

8. For top level events only, tick the 'Assign a New Guid Automatically if None Provided' checkbox.

---

Note: Subevents must have 'Assign a New Guid Automatically if None Provided' un-ticked when running Multi-Site Sync

---

To configure a Bxflid parameter for all events:

1. Open the Configurator.
2. Click on the **Import / Export** button.
3. Click on the **Run Import Extensions** button.
4. Click on the 'BXF' tab.
5. Click on the **Add Bxflid Parameter for Every Event Type** button (Figure 2-5).

---

Note: When adding a Bxflid to all events in this manner, the system will automatically enable 'Assign a New Guid Automatically if None Provided' - when running Multi-Site Sync, subevents must have this parameter disabled. Follow the instructions for configuring a Bxflid parameter for a single event type, above.

---

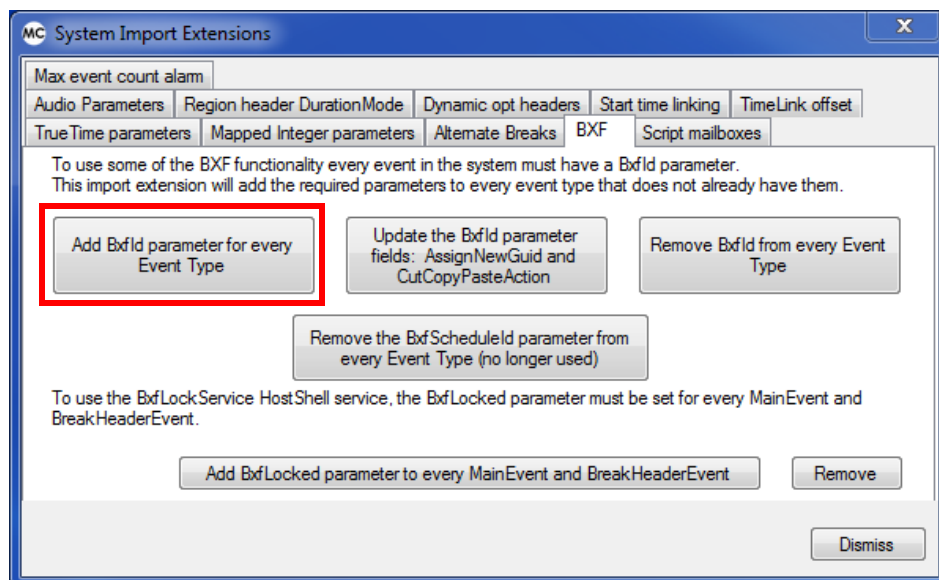


Fig. 2-5: Adding a Bxflid Parameter for Every Event Type

The Bxflid is a required event parameter for the following BXF services:

- BXF Query Service
- BXF Schedule Service
- BXF Schedule Change Notifier Service

---

Note: The BXF standard specifies the element name for a Bxflid as <EventId>. As the name 'EventId' is already used in Morpheus it had to be changed to 'Bxflid'.

---

## Translations

The BXF standard defines a number of default event types that must be matched to specific Morpheus event types prior to being loaded into the EventStore. The matching process is known as a 'Translation'.

For inbound messages: match the BXF event types received in the BXF message to Morpheus event types.

For outbound messages: the contents of the message requires no translation apart from the event type information extracted from the EventStore, which must be translated to the BXF standard event type.

## MAPP Process IDs (PIDs) and PID Chains

The PID is a unique number that is applied to a MAPP service (or service instance) as an identifier - no two MAPP services or service instances should be configured with the same PID. It is used by MAPP services and service instances only in order to identify one from another within the MAPP environment, and is not therefore included in any BXF message.

The only BXF service that is identified by a PID is the 'MAPP BXF Transfer Request Service' (refer to MAPP BXF Transfer Request Service).

PID chains are created when a MAPP service, that is targeted to process a message, is configured to reference another service; for example, if the original target service is unable to perform the requested processing. The second service may also be configured to reference a third service, and so on.

---

Note: The creation of a PID loop, where the last service in the chain references the first, is not recommended

---

## Allocating Process IDs (PIDs)

A PID is only used to identify a MAPP service, and should be allocated leaving gaps in the numbering system in order to allow for expansion. The recommendation is to allocate a PID in increments of 5; i.e. starting at number 5 for the first service, then 10, 15, 20 etc.

As the 'MAPP BXF Transfer Request Service' is the only BXF service that is identified by a PID, it is recommended that its number is distanced from those of the MAPP services, for example, allocated the number 100.



## BXF Dates and Times

The BXF protocol supports the following UTC date and time format only:

- yyyy-mm-ddThh:mm:ss.msZ (the Z suffix indicates that the given date and time is UTC)

### Example

2013-10-10T09:00:00.100Z

## BXF Schema Versions

Morpheus supports the following SMPTE 211 BXF schema versions:

- v1.0 2008
- v2.0 2012
- v3.0 2013

For those BXF services that generate unsolicited messages, the schema version must be specified, such as the BXF AsRun Notifier service as shown in the example below:

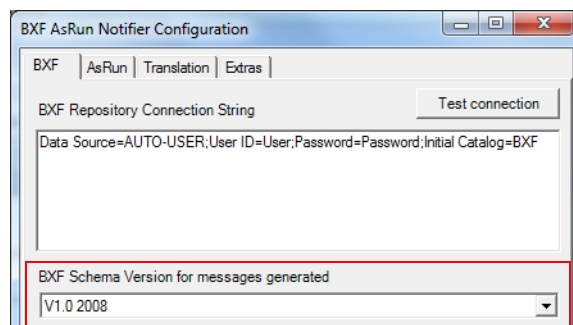


Fig. 2-6: Example BXF Specification Configuration

Those services that are solicited for information will do so according to the schema version of the received message.

## BXF Host Shell Services

This section briefly describes the characteristics of the available BXF Host Shell Services.

- **BXF AsRun Notifier Service** - queries for new records in the AsRun database for transmission in a BXF message to another system. Refer to BXF AsRun Notifier Service.
- **BXF Content Transfer To Record Service** - generates Morpheus events from a received 'BXF Content Transfer' message in order for a recording to be performed in Morpheus. Refer to BXF Content Transfer To Record Service.
- **BXF Database Maintenance Service** - a housekeeping service that deletes old BXF message entries from the BXF database. Refer to BXF Database Maintenance Service.
- **BXF File Reader Service** - monitors a watch folder for received file-based BXF messages that are processed according to the configuration of the service. Refer to BXF File Reader Service.
- **BXF File Writer Service** - queries the BXF database for messages with destination folders that it is configured for, transmitting them as files to another system. Refer to BXF File Writer Service.
- **BXF Import Content Service** - receives BXF Data messages from an external system with actions that must be applied to content entries in the MAPP database. Refer to BXF Import Content Service
- **BXF Lock Service** - protects events that are close to going on air against schedule updates contained in received BXF messages. Refer to BXF Lock Service.
- **BXF Media Change Notifier Service** - queries tables in the MAPP database in order to retrieve any changes to material and generates a BXF content change message for posting to an external system. Refer to BXF Media Change Notifier Service.
- **BXF Multiplex Message Service** - duplicates a BXF Query message for material that has been received from an external source into a core BXF system, for transmission to individual remote systems for processing. Refer to BXF Multiplex Message Service.
- **BXF Query Service** - processes BXF Query Request messages and interrogates the MAPP database or the Morpheus EventStore for the required information. Refer to BXF Query Service.
- **BXF Schedule Change Notifier Service** - upon receipt of schedule change notifications from the EventStore, this service will generate a BXF message for transmission to preconfigured destinations. Refer to BXF Schedule Change Notifier Service.
- **BXF Schedule Service** - receives schedules in a BXF format from which it will update the EventStore. Refer to BXF Schedule Service.
- **BXFSocket Reader Service** - reads BXF messages transmitted by one or more BXF Socket Writer services, or equivalent services from a third party external system. Refer to BXF Socket Reader Service.
- **BXF Socket Writer Service** - queries the BXF database for messages with destination services that it is configured for, transmitting them as IP packets. Refer to BXF Socket Writer Service.
- **MAPP BXF Transfer Request Service** - a MAPP agent that transacts 'BXF Content Transfer Requests', processes BXF Query messages, and processes material transfer update requests from an external system. Refer to MAPP BXF Transfer Request Service.

## BXF Ardome SOAP Sender Service

---

Note: This service has been deprecated.

---

## BXF AsRun Notifier Service

The BXF AsRun Notifier service performs the following tasks:

1. Connects to the configured AsRun database for the configured channel.
2. Queries the existence of any new AsRun Records.
3. Translates the new AsRun records into a BXF message format.
4. Transfers the translated message to the BXF database.

By design, this service queries AsRun records in order to generate a message for a complete schedule only - this default behaviour can be modified using the - **Time Waiting to Assume Schedule Play Out Completed...** feature (refer to AsRun Tab).

---

Note: An instance of this service is dedicated to the one channel that it is configured for.

---

---

Note: A Morpheus BXF 'Notifier' service is only responsible for retrieving messages from a configured location and depositing them in the BXF database for collection by a BXF 'Writer' service.

---

### BXF AsRun Notifier Service: Principle of Operation

The functionality of the BXF AsRun Notifier service is shown in Figure 2-7 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

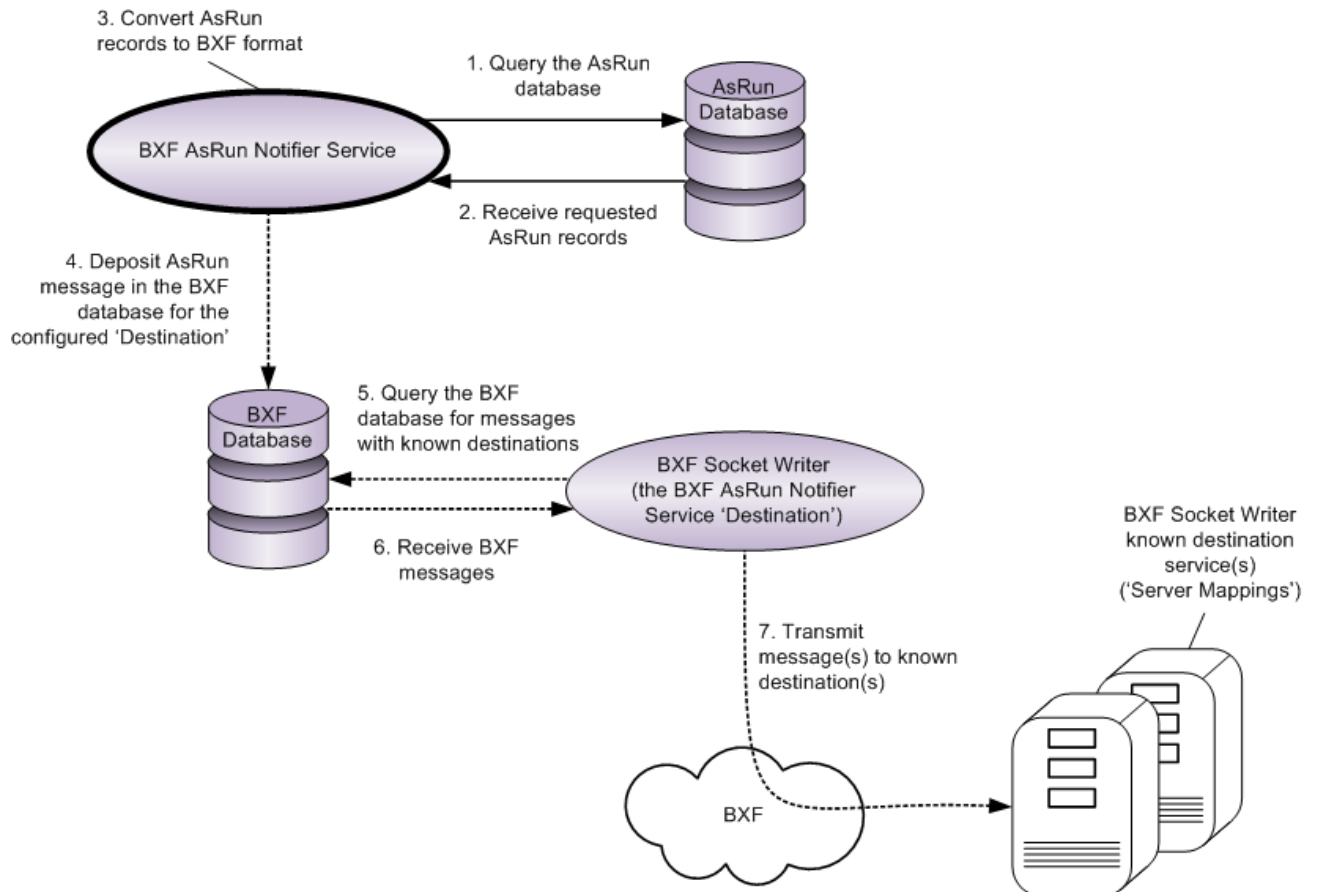
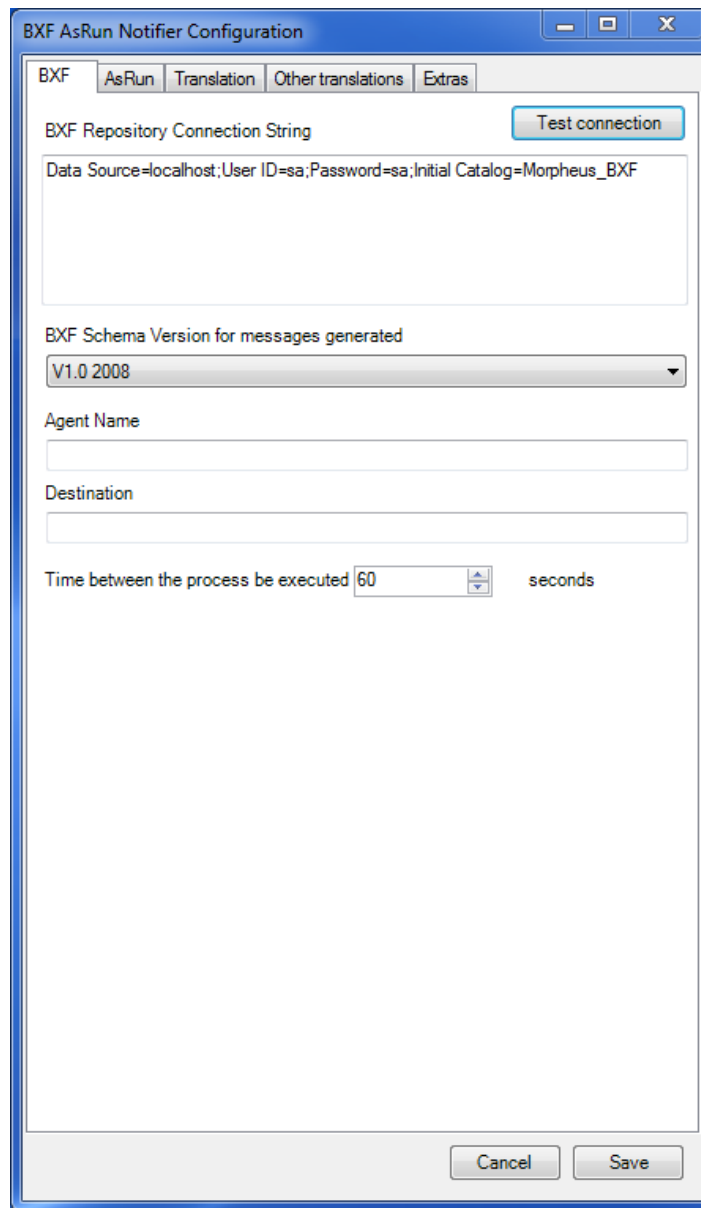


Fig. 2-7: BXF AsRun Notifier Service Logical Topology

## BXF Tab

BXF specific parameters.



The image shows a Windows-style configuration window titled "BXF AsRun Notifier Configuration". It has a tabbed interface with four tabs: "BXF", "AsRun", "Translation", and "Other translations". The "BXF" tab is currently selected. Inside the window, there is a "BXF Repository Connection String" section with a text box containing "Data Source=localhost;User ID=sa;Password=sa;Initial Catalog=Morpheus\_BXF" and a "Test connection" button. Below this is a "BXF Schema Version for messages generated" section with a dropdown menu showing "V1.0 2008". There are also text boxes for "Agent Name" and "Destination". At the bottom, there is a "Time between the process be executed" section with a spinner box set to "60" and the unit "seconds". At the very bottom of the window are "Cancel" and "Save" buttons.

BXF AsRun Notifier Configuration

BXF AsRun Translation Other translations Extras

BXF Repository Connection String Test connection

Data Source=localhost;User ID=sa;Password=sa;Initial Catalog=Morpheus\_BXF

BXF Schema Version for messages generated

V1.0 2008

Agent Name

Destination

Time between the process be executed 60 seconds

Cancel Save

Fig. 2-8: BXF AsRun Notifier Service Configuration Window

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxfl catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### - Agent Name

Enter a name that will identify the service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the BXF database) when sending a BXF message, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages.

---

Note: It is recommended that the agent name is different to the instance name.

---

**- Destination**

A BXF 'Writer' service will be responsible for delivering messages, that have been deposited into the BXF database by this service, to its destination.

Enter the name of the destination service to which a BXF 'Writer' service will deliver messages.

---

Note: If no 'Writer' service has been configured, messages will remain on the database until removed by the 'BXF Database Maintenance Service'.

---

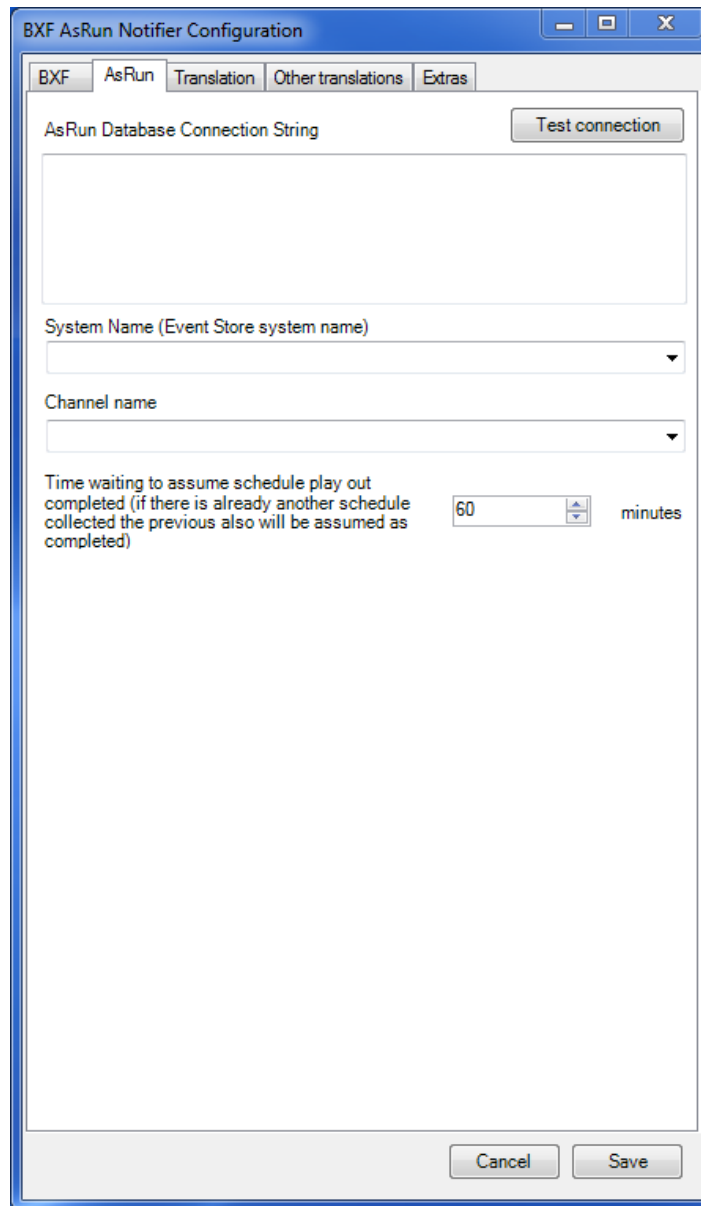
**- Time Between the Process be Executed**

The frequency at which this service will inspect the AsRun database for the existence of new records.

Enter a frequency in seconds. The default is 60.

## AsRun Tab

The criteria required in order for this service to query specific AsRun records.



The image shows a Windows-style configuration window titled "BXF AsRun Notifier Configuration". It has a tabbed interface with five tabs: "BXF", "AsRun", "Translation", "Other translations", and "Extras". The "AsRun" tab is currently selected. Inside the window, there is a section for "AsRun Database Connection String" with a large text input field and a "Test connection" button. Below this, there are two dropdown menus: "System Name (Event Store system name)" and "Channel name". Further down, there is a label "Time waiting to assume schedule play out completed (if there is already another schedule collected the previous also will be assumed as completed)" followed by a spin box set to "60" and the unit "minutes". At the bottom right, there are "Cancel" and "Save" buttons.

Fig. 2-9: BXF AsRun Notifier Service Configuration - AsRun Tab



### - AsRun Database Connection String

Enter the connection string that will provide access to the AsRun database that this service will query for the existence of new records. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwr>;Initial Catalog=<bx catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalog=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - System Name (Event Store System Name)

From the drop-down menu, select a Morpheus system (EventStore). This service will query the AsRun database of the selected system.

### - Channel Name

From the drop-down menu, select a channel. This service will query the AsRun database for events from the selected channel only.

### **- Time Waiting to Assume Schedule Play Out Completed...**

The BXF AsRun Notifier Service is designed to only generate a message that contains a whole schedule (based upon a single Schedule Information Object [SIO]). The standard behaviour, which will still apply with this feature configured, is such that at the point at which the service detects that one or more events in a different schedule are present in the AsRun database (as indicated by a different SIO), it concludes that the previous schedule must have completed and will therefore generate a related message.

Given the situation where an event is on hold, the service would have to wait until the held event has completed before generating a message, potentially creating long delays. This feature overcomes such a predicament by generating a message for all events prior to the held event that are in the same schedule once a configured period of time has expired. The clock resets at the end of each event.

The process will therefore perform two separate actions:

- Generate a message at the end of a schedule identified by a single SIO
- Generate a message for all previous events in a schedule (identified by a single SIO) if the current event overruns for more than the time set for this feature.

Specify a time limit for any single event to appear in the AsRun database. The default is 60 minutes. This feature cannot be disabled.

---

Note: The behaviour of this feature is not limited to held events - it will apply to any event with a duration that exceeds the configured limit.

---

## Translation Tab

The Morpheus event types from the AsRun database must be matched to those defined in the BXF standard for inclusion in the messages that this service will generate. The matching process is known as a 'Translation'.

**BXF AsRun Notifier Configuration**

AsRun Type: BasicAsRun

Translate BXF Primary event to: [Empty dropdown]

Translate BXF Primary-ProgramHeader event to: Not supported - refer to parameter description

Translate BXF Primary-BreakHeader event to: [Empty dropdown]

Translate BXF Comment event to: Not supported - refer to parameter description

Translate BXF Macro event to: Not supported - refer to parameter description

Translate NonPrimary event to:

NonPrimaryEventName	Morpheus EventType

A: [Empty text field] B: [Empty dropdown]

Add Replace Delete Export NonPrimary event

Cancel Save

Fig. 2-10: BXF AsRun Notifier Service Configuration - Translation Tab

### - AsRun Type

Select the depth of event information from the AsRun database that will be translated to BXF standard event types, and then included in the message:

---

Note: If it is intended that event parameter fields are to be inserted into the 'PrivateInformation' element of the BXF messages generated by this service, then select **Complete AsRun**, and refer to the Extras Tab.

---

- **Complete AsRun**  
Verbose.
- **Basic AsRun**  
Concise. This is the default.

### - Translate BXF Primary Event To

No default.

From the drop-down menu select the Morpheus event type that will translate to the BXF Primary Event type.

### - Translate BXF Primary-ProgramHeader Event To

This BXF event type is not currently supported by the BXF AsRun Notifier Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

### - Translate BXF Primary-BreakHeader Event To

No default.

From the drop-down menu select the Morpheus event type that will translate to the BXF Primary-BreakHeader Event.

### - Translate BXF Comment Event To

This BXF event type is not currently supported by the BXF AsRun Notifier Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

### - Translate BXF Macro Event To

This BXF event type is not currently supported by the BXF AsRun Notifier Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

### - Translate NonPrimary Event To

No defaults.

Non-primary event names (e.g. sub-events for GPIs and subtitles) are not specified in the BXF standard, and as such they are user definable.

---

Note: Any NonPrimaryEventName configured for this feature must match with that configured on the system from which messages will be received and responses transmitted.

---

- **Add**

- a) In the text box 'A', enter a name for the non-primary event that will be sent in the BXF message.

---

Note: Names are case sensitive.

---

- b) From the drop-down menu 'B', select the Morpheus event type to translate to the non-primary event selected in box 'A'.
- c) Click on the **Add** button - the new translation entry is added to the table.

- **Replace**

To modify an existing translation entry, click on any part of it and the text box and drop-down menus will be populated with its current properties. Enter alternative properties then click on the **Replace** button.

- **Delete**

To remove a translation entry from the table, click on any part of it and click on the **Delete** button.

### - **Export NonPrimary Event**

Save to a file the non-primary event translations configured for a selected channel. This function is only provided for administrative records - there is no import capability to allow sharing of the configured translations.

1. From the channel list, select the one for which to export the translations.
2. Click on the **Export NonPrimary Event** button.
3. Windows Explorer is opened: enter a filename and destination folder, then click on the **Save** button

## Other Translations Tab

The BXF standard specifies three video / audio event transition rates: 'Fast', 'Medium', and 'Slow' - these are a representation of actual frame rates. Morpheus, however, processes actual frame rates only, and does not recognise the BXF names. In order for Morpheus to be compatible with the BXF standard, a translation mechanism is accessible from the 'Other Translations' configuration tab.

This service uses the translation mechanism in order to generate messages.

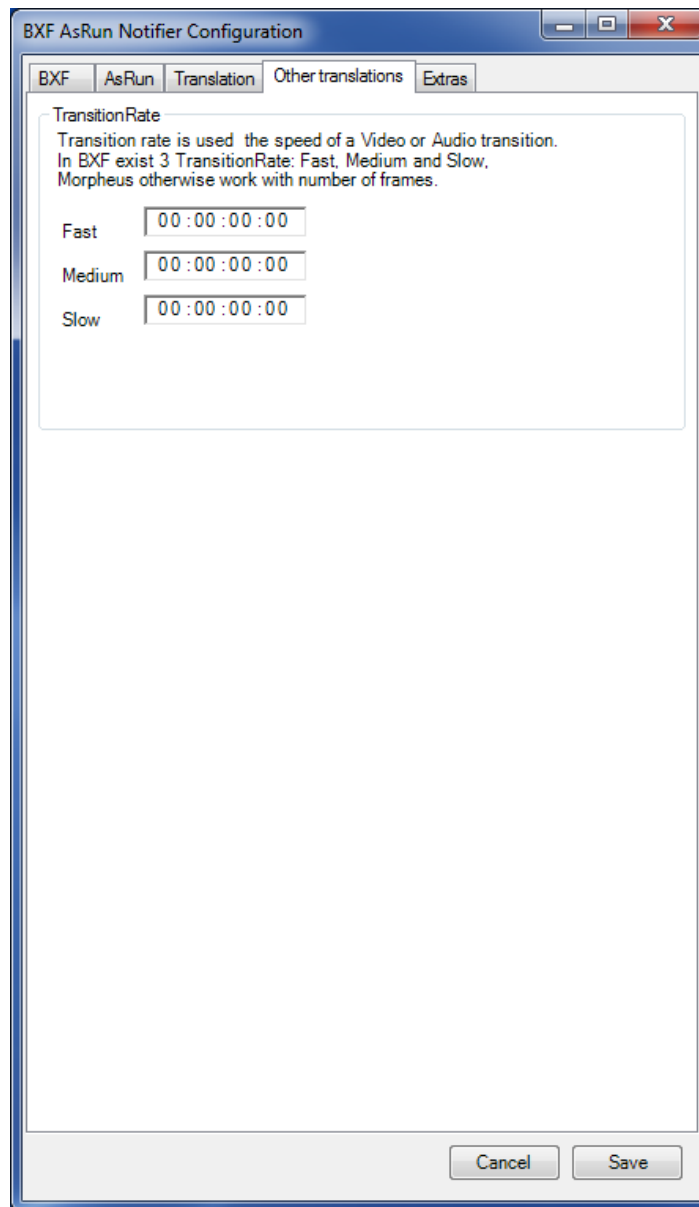


Fig. 2-11: BXF AsRun Notifier Service Configuration - Other Translations Tab

### Transition Rate

#### **- Fast / Medium / Slow**

The TransitionRate fields allow specific Morpheus frame rates to be translated to the BXF named transition rates for transmission in a BXF message.

## Extras Tab

The BXF standard defines a 'PrivateInformation' element for a message, into which any form of information can be inserted for transfer to another application. The purpose is to provide the means of transferring metadata within a standard BXF message that does not comply with the BXF schema format.

The 'Extras' tab provides the means of specifying event parameters, the properties of which are extracted from the AsRun, and inserted into the 'PrivateInformation' element of the BXF message.

---

Note: In order to successfully use this function, **Complete AsRun** must be selected as the 'AsRun Type' (refer to the Translation Tab).

---

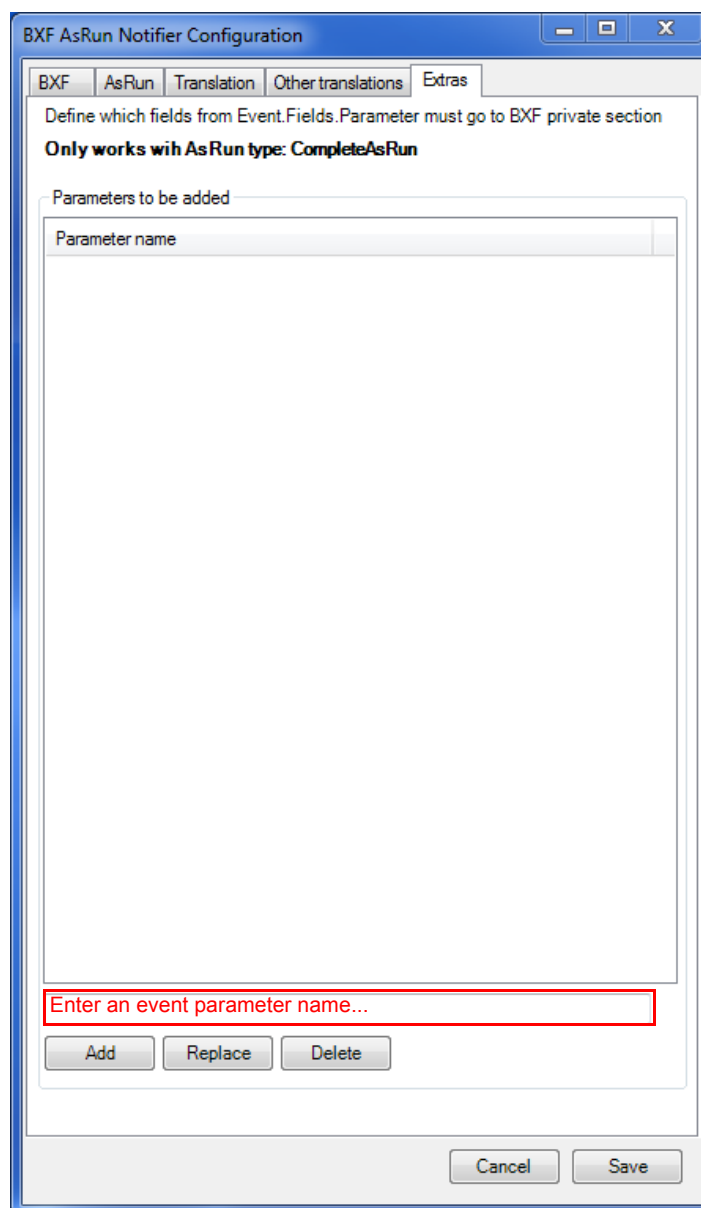


Fig. 2-12: BXF AsRun Notifier Service Configuration - Extras Tab



### Parameters to be Added

- **Add**

- a) Enter the event parameter name.

---

Note: Parameters are case sensitive.

---

- b) Click on **Add**. A warning message is displayed if the parameter name is already configured for this instance of the service.

- **Replace**

Highlight the event parameter to be changed - the text entry boxes are populated with the existing entry. Enter the new parameter name and click on **Replace**.

- **Delete**

Highlight the event parameter(s) to delete then click on **Delete**.

---

Note: To view a list of parameter definitions for a particular event, follow the instructions below:

1. In the Configurator application, click on the **Configure Events** button
  2. Select an event.
  3. Click on the 'All Parameter Definitions' tab.
- 



## BXF Content Transfer To Record Service

A service that generates Morpheus events from a received 'BXF Content Transfer' message in order for a recording to be performed. A single 'BXF Content Transfer' message may contain one or more record requests, each of which specifies information for a record event and a pad event that will stop the recording and bridge the schedule until the next main event (Morpheus does not permit gaps between main events). A separate schedule is created for each record request received in a message.

---

Note: This is the only service that will process messages with a 'Transfer Type' of *Recording*.

---

The following properties are supplied for the record event:

- Inpoint
- Duration
- Material Id
- Target Device Id (the device on which to perform the recording)

The following properties are supplied for the pad event which is a follow-on event that will run after the record event:

- Duration = 1 second

## BXF Content Transfer to Record Service: Principle of Operation

The functionality of the BXF Content Transfer to Record service is shown in Figure 2-13 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

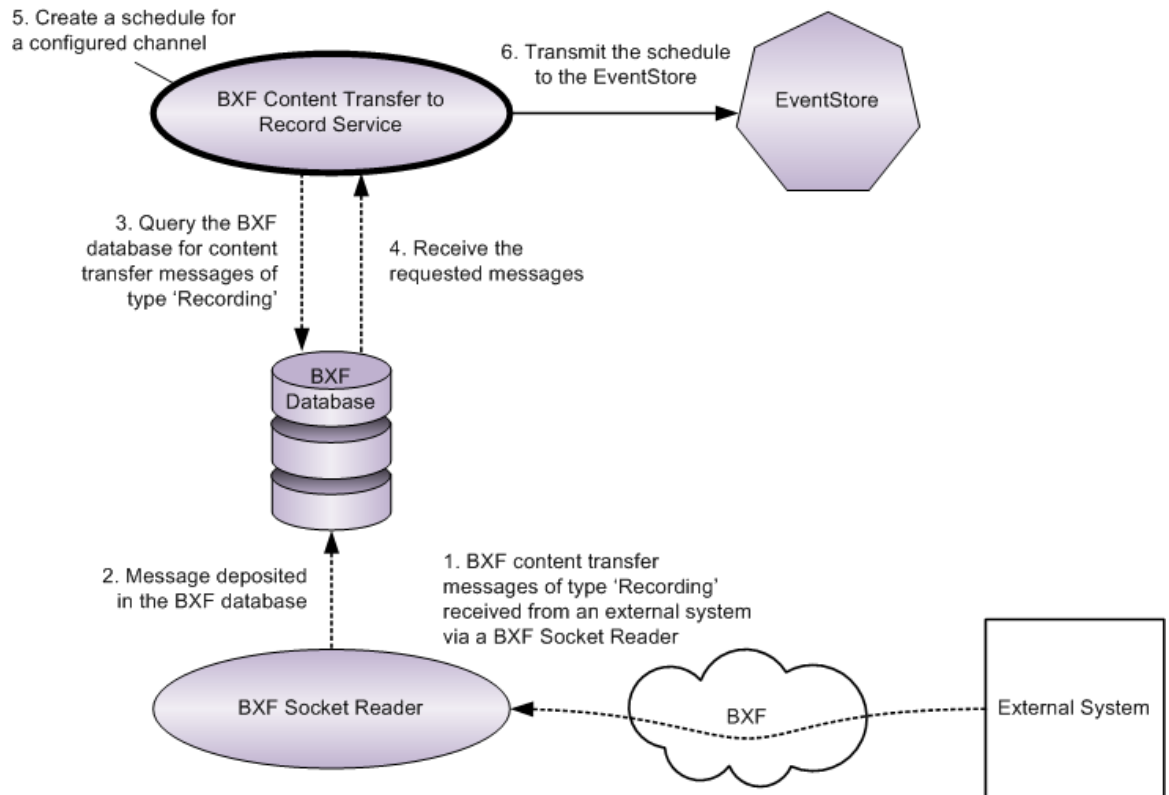


Fig. 2-13: BXF Content Transfer to Record Service Logical Topology

## BXF Content Transfer To Record Service Configuration Parameters

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwr>;Initial Catalog=<bxr catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

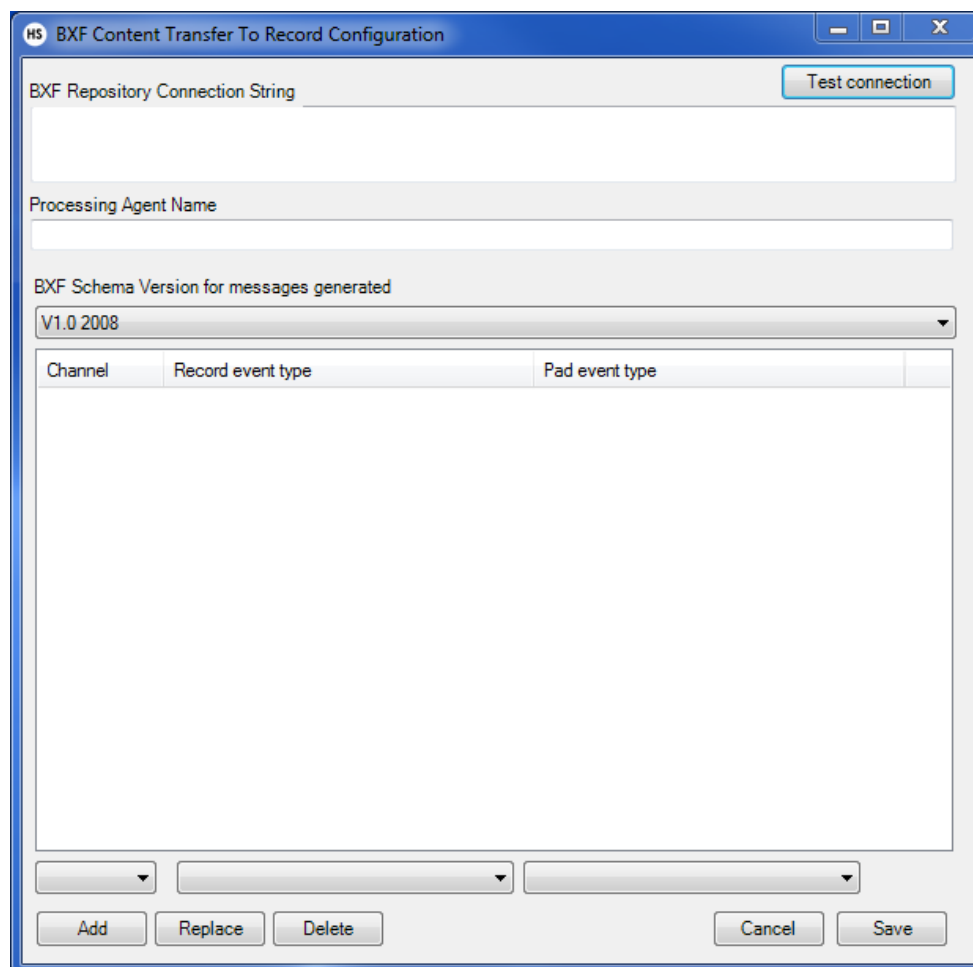


Fig. 2-14: BXF Content Transfer to Record Service Configuration Window

### - Processing Agent Name

Enter a name that will identify the BXF Content Transfer to Record Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Processing Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message to the external system, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages from the external system.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---



---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### Adding Channels to Perform the Record Events

As channels are not specified in the received BXF message, it is necessary to make a selection in the configuration of this service from those available in the Morpheus system. Upon receiving a BXF message of type 'Recording' the service will generate a schedule using the following principles:

1. Starting from the top of the list of configured channels, the first one that is available to perform a recording for its full duration will be selected. The availability of a channel is based upon whether or not it has previously scheduled events that will clash with the recording request.
2. Apply the configured record event type.

---

Note: The configured 'Record Event Type' must be a main event with an existing 'record' sub-event.

---

3. Apply the configured pad event. The pad event will run in-between events in order to fill gaps in the schedule, for example at the point at which a record event has completed and there are no following events

---

Note: Morpheus does not allow gaps in schedules.

---

4. This service will schedule any subsequent recording requests for the same channel, unless there is a clash with an existing event.

---

Note: If no channels are available when a BXF Content Transfer request is received, then a NAK will be returned to the requesting service.

---

### - Add / Replace / Delete

---

Note: All values are case sensitive.

---

- **Add**

- a) From the drop-down menu, select a channel upon which a recording can be scheduled
- b) From the drop-down menu, select a Record Event Type for the channel (a main event that is already configured with a 'record' sub-event)

**IMPORTANT:** The BXF message does not support the designation of a source - it must therefore be specified as a default in the selected 'Record Event Type' for this service.

- c) From the drop-down menu, select a pad event for the channel

---

Note: A single channel can only be added to the list once.

---

- **Replace**

Highlight the channel entry to be changed - the text entry boxes are populated with the existing entries. Select an alternative 'Record Event Type' and / or 'Pad Event Type' then click on **Replace**.

- **Delete**

Highlight the channel entry to be deleted then click on **Delete**.

---

Note: The drop-down menus are populated from the local EventStore.

---

## BXF Database Maintenance Service

A housekeeping service that examines the BXF database every 60 minutes, deleting aged messages.

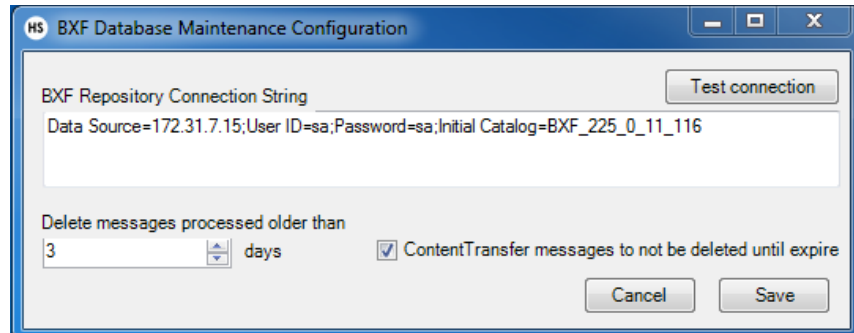


Fig. 2-15: BXF Database Maintenance Service Configuration Window

### BXF Database Maintenance Service: Principle of Operation

The functionality of the BXF Database Maintenance service is shown in Figure 2-16.

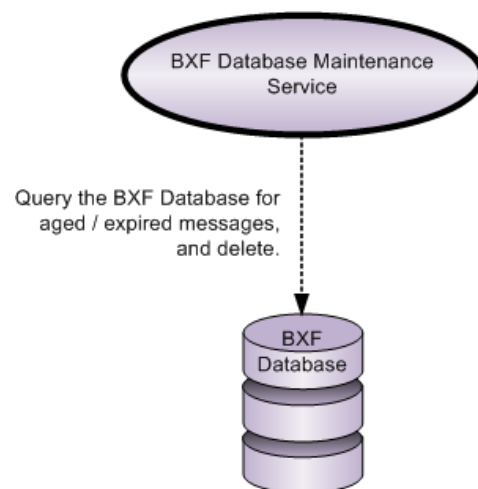


Fig. 2-16: BXF Database Maintenance Service Logical Topology

## BXF Database Maintenance Service Configuration Parameters

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxfl catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Delete Messages Processed Older Than <n> Days

Default: 3 days.

Delete any message, processed or not, from the BXF database once the value attributed to this parameter has been exceeded.

---

WARNING: If given a value of 0 (zero) then all messages will be deleted from the database!

---

### - Content Transfer Messages to not be Deleted Until Expire

Default: enabled (ticked).

Applies only to content transfer messages.

When enabled, content transfer messages are deleted according to their embedded expiry date, thus overriding the 'Delete Messages Processed Older Than <n> Days' feature.

If disabled, content transfer messages are treated in the same manner as any other message type.



## BXF File Reader Service

The BXF File Reader service monitors a watch folder for received file-based BXF messages. Messages are parsed, validated against the BXF schema, and processed according to the configuration of the service as follows:

- Deposit in the BXF database
- Move / rename / delete the received file

---

Note: A BXF File Writer service does not transmit messages directly to a BXF File Reader service.

---

### BXF File Reader Service: Principle of Operation

The functionality of the BXF File Reader service is shown in Figure 2-17 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

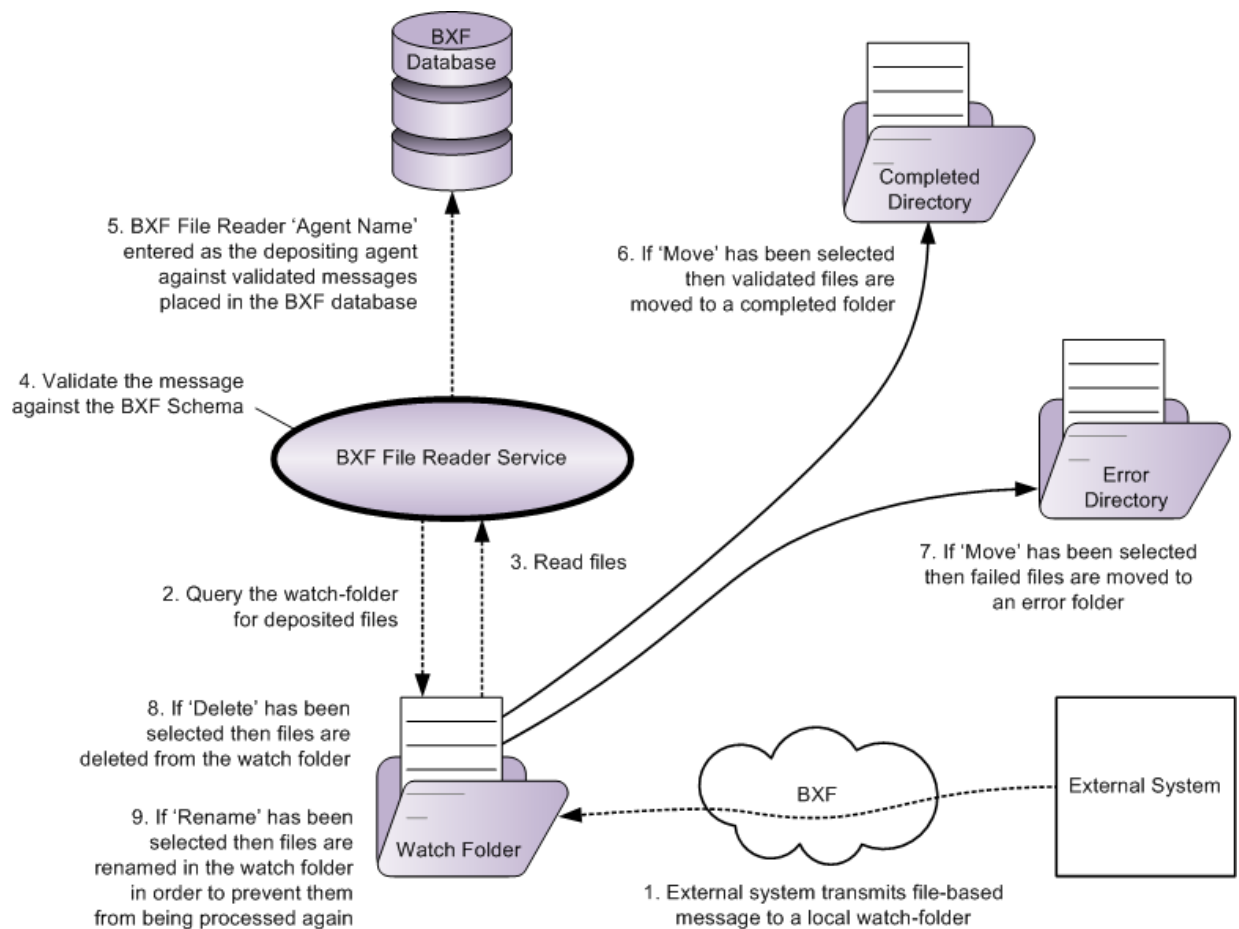


Fig. 2-17: BXF File Reader Service Logical Topology

## BXF File Reader Service Configuration Parameters

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwr>;Initial Catalog=<bxr catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

Note: The properties for this string will have been configured at the time of the database creation.

Verify the validity of the connection string by pressing the **Test Connection** button.

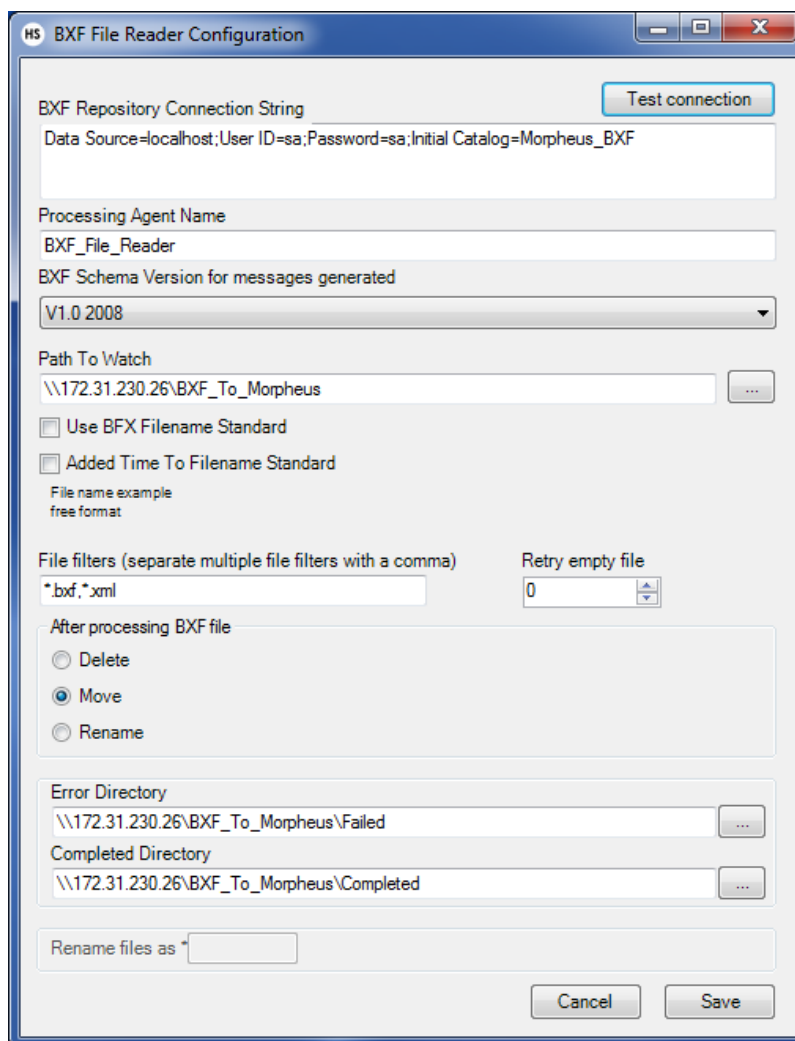


Fig. 2-18: BXF File Reader Configuration

### - Processing Agent Name

Enter a name that will identify the BXF File Reader Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). Exceptionally, the Agent Name attributed to a BXF File Reader service is only used to record its identity in the BXF database as the agent responsible for depositing a particular message. The Agent Name of a BXF File Reader service is never used as addressing information by any other service.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---


---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### - Path To Watch

BXF messages are received into a folder in order to await processing by this service.

Use the ellipsis button  in order to select the path for the watch folder that this service will monitor for received messages.

### - Use BXF Filename Standard

Default: disabled (un-ticked).

When ticked, it indicates to this service that any file deposited in the watch folder adheres to the recommended file-naming strategy defined in the BXF standard. For example:

BXF[20091130][BXF File Reader][ContentShare][0a6842f3-08f6-488b-b802-f01c249d96ed].xml

---

Note: A NAK can only be sent to the originator of a message if the BXF standard file-naming convention is used, for the following reason:

---

Should the contents of a message be invalidated, the only way in which the service can identify the originator is through the filename.

---

#### - Added Time to Filename Standard

Default: disabled (un-ticked).

This checkbox is only available if the **Use BXF Filename Standard** checkbox (above) is ticked.

When this feature is enabled, the time at which the file was created is appended to the date portion of the filename. The format of the time is **hhmmss.xxxx** (where **xxxx** represents milliseconds). This is a proprietary extension to the BXF file-naming standard.

##### Example

BXF[20091130\_125209.6362][BXF File Reader][ContentShare]  
[0a6842f3-08f6-488b-b802-f01c249d96ed].xml

---

Note: The **Added Time To Filename Standard** naming option is not defined within the BXF standard, however it allows the processing of messages in date & time order instead of by date order only.

---

#### - File Filters

Define the file types that this service will be limited to processing - specify the file types required as Comma Separated Values (CSV).

Default: \*.bxf, \*.xml

---

Note: This option is only available if **Use BXF Filename Standard** is disabled.

The **File filters** option does not adhere to the BXF standard - no notifications will be issued for file types that have been filtered out, and will therefore remain unprocessed.

---

#### - Retry Empty File

Default = 0.

An empty file may appear in the watch folder for processing, indicating that the originating service has yet to write to the file.

Enter a value to define the number of times that this instance of the service will attempt to read the contents of the file - retries will occur once every second.

### - After Processing BXF File: Delete / Move / Rename

Default: **Move**.

The BXF Reader Service can be configured to delete, move or rename files after processing.

Using the radio buttons, select one action from the following:

- **Delete**: delete the file after validation and processing or following a validation failure.
- **Move**: move the file according to the following rules.
  - After successful validation, processing, and depositing in the BXF database, move the file to the folder specified for **Completed Directory**, below.
  - If a failure occurs during validation, move the file to the folder specified for **Error Directory**, below.
- **Rename**: if required for administrative purposes, append an additional extension to the filename once processing is complete, as specified in **Rename File As**, below. Once renamed, the file remains in the watch folder but will not be processed again.

### - Error Directory

If the **Move** action has been selected (above), and should a failure occur during validation, then the file is moved to the folder specified here.

Use the ellipsis button  to select a folder.

### - Completed Directory

If the **Move** action has been selected (above), and the message has successfully validated, then the file is moved to the folder specified here.

Use the ellipsis  button to select a folder.

### - Rename Files As

If the **Rename** action has been selected (above), then append the extension specified here to the original filename. Once renamed, the file will not be processed again.

Enter any extension of choice, preceded by a dot.

## BXF File Writer Service

The BXF File Writer service queries the BXF database for messages with destinations that it is configured for, transmitting them as files.

---

Note: The BXF File Writer only sends a message to the named destination.

---

---

Note: A BXF File Writer service does not transmit messages directly to a BXF File Reader service.

---

### BXF File Writer Service: Principle of Operation

The functionality of the BXF File Writer service is shown in Figure 2-19 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

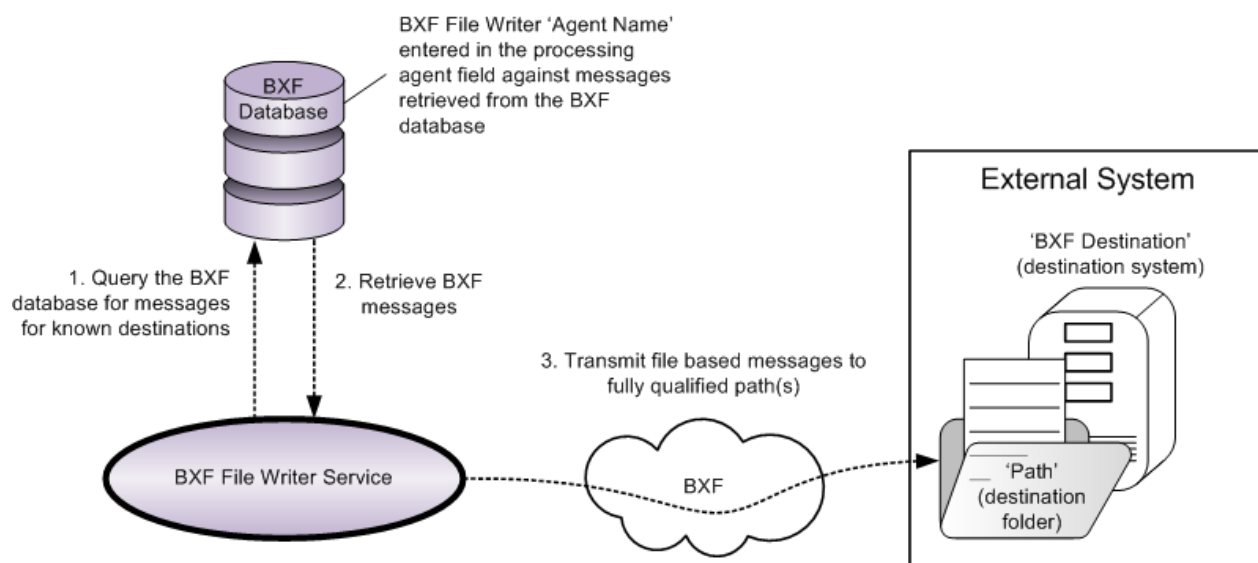


Fig. 2-19: BXF File Writer Service Logical Topology

## General Settings Tab

Fig. 2-20: BXF File Writer Service Configuration - General Settings Tab

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxg catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Agent Name

Enter a name that will identify the service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). Exceptionally, the Agent Name attributed to a BXF File Writer service is only used to record its identity in the BXF database as the agent responsible for retrieving a particular message. The Agent Name of a BXF File Writer service is never used as addressing information by any other service.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - Required Acknowledge

Default: disabled.

When enabled, this instance of the service will resend the message unless it has received an ACK from the destination service.

The number of retries and their frequency is configured in **Number of Retries** and **Time Delay Between Each Retry**, below.

### - Number of Retries

Default: 0

If **Required Acknowledge** (above) is enabled, then enter the number of times that this instance of the service will retry to send the same message upon failing to receive an ACK from the destination service.

The recommended value is 3.

### - Time Between Each Retry

Default: 0

If **Required Acknowledge** (above) is enabled, then enter the frequency at which this instance of the service will retry to send the same message upon failing to receive an ACK from the destination service.

Enter a time in seconds - the recommended value is 5.

### - Multi Thread

Default: disabled.

By default, an instance of this service opens a session with only one destination for the duration of a message transmission and the receipt of an ACK.

Enable this function in order for this instance of the service to conduct multiple session transactions concurrently with the destinations configured on the 'Server Mappings' tab.

---

Note: There is no known limitation to the number of possible concurrent threads.

---



### Add Extra Information to the Filename...

The standard file-naming convention adheres to the BXF specification, with proprietary options to append additional information that relates to the message.

#### **- Add Time Together With Date**

Default: disabled.

The time at which the file was created is appended to the date portion of the filename, allowing the destination service to process messages in date / time order.

The format of the time is **hhmmss.xxxx** (where **xxxx** represents milliseconds).

A representation of the selected structure of the filename is displayed dynamically along the bottom of the window, as the checkboxes are ticked.

#### **Example**

- **BXF standard**

```
BXF[20091130] [BXF File Reader] [ContentShare]
[0a6842f3-08f6-488b-b802-f01c249d96ed] .xml
```

- **BXF standard with time**

```
BXF[20091130 125209.6362] [BXF File Reader] [ContentShare]
[0a6842f3-08f6-488b-b802-f01c249d96ed] .xml
```

#### **- Add Origin MessageId**

Default: disabled.

Inserts the *originMessageId* before the date in the file name. The *originMessageId* is a Universally Unique Identifier (UUID) used in response messages that refers to the origin message ID.

A representation of the selected structure of the filename is displayed dynamically along the bottom of the window, as the checkboxes are ticked.

#### **Example**

```
BXF[51f043ae-f735-4f48-ac2e-719852c838df] [20091130] [BXF File
Reader] [ContentShare] [0a6842f3-08f6-488b-b802-f01c249d96ed] .xml
```

## Server Mappings Tab

Define the destination systems ('BXF Destinations') to which this BXF File Writer Service will transmit messages.

Note: Messages to be retrieved by this service are deposited in the BXF database addressed to a destination system - this service identifies those messages in the database that it must transmit by using the destination systems that is configured with.

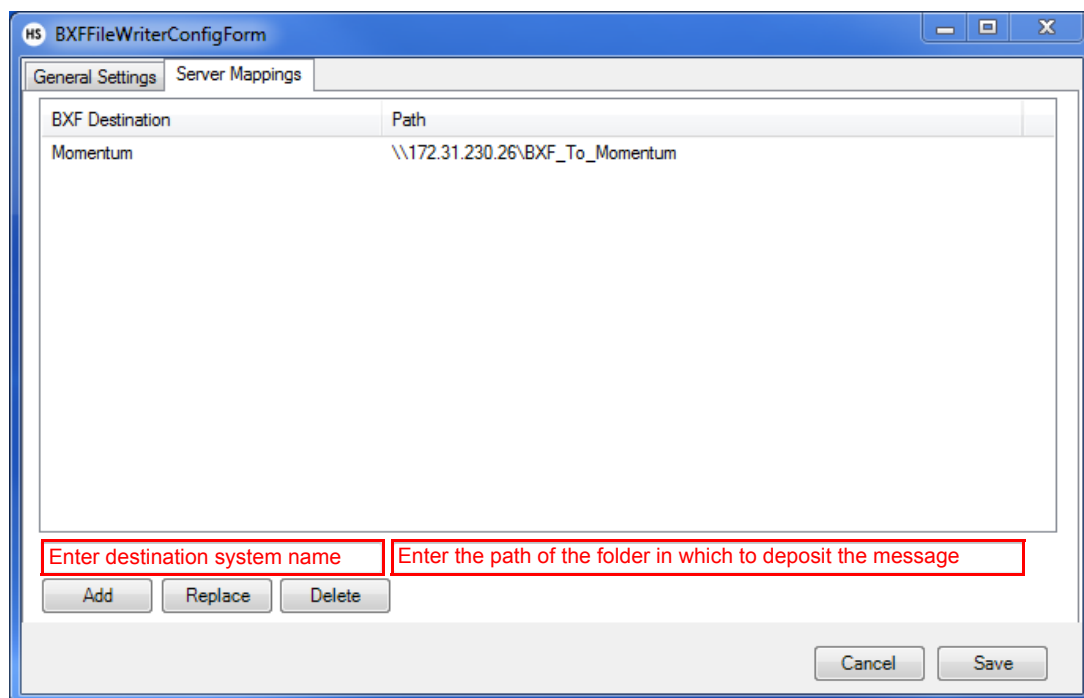


Fig. 2-21: BXF File Writer Service Configuration Window - Server Mappings Tab

## - Add / Replace / Delete

Refer to Figure 2-21.

- **Add**

- a) Enter the name of the destination system to which messages are to be sent (i.e. the name of the external system).

---

Note: Parameters are case sensitive.

---

- b) Enter the fully qualified path for the folder in which to deposit the message.

- c) Click on **Add**. The system name and path will appear in the main pane.

---

Note: A path can be expressed as a local file system or according to the Uniform Naming Convention (UNC).

Examples

D:\<destination>

\\<IP address>\destination

\\<server>\<volume>\<destination>

---

- **Replace**

Highlight the destination entry to be changed - the text entry boxes will be populated with the entry. Enter the new destination properties in the appropriate text boxes, click on **Replace**.

- **Delete**

Highlight the destination entry to delete then click on **Delete**. This function deletes the whole destination entry.

## BXF Import Content Service

A service that receives BXF Data messages from an external system with the actions *add*, *update*, and *remove*, that must be applied to content entries in the MAPP database. This service has no further responsibilities other than sending a response to the external system from which a message was received.

---

Note: This service only validates the format of the received message against the configured schema, not the accuracy of the content.

---

---

Note: It is recommended that received messages only contain a single clip for processing by this service, thereby allowing for an ACK or NAK response to reference individual rather than assorted content.

---

### BXF Import Content Service: Principle of Operation

The functionality of the BXF Import Content service is shown in Figure 2-22 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

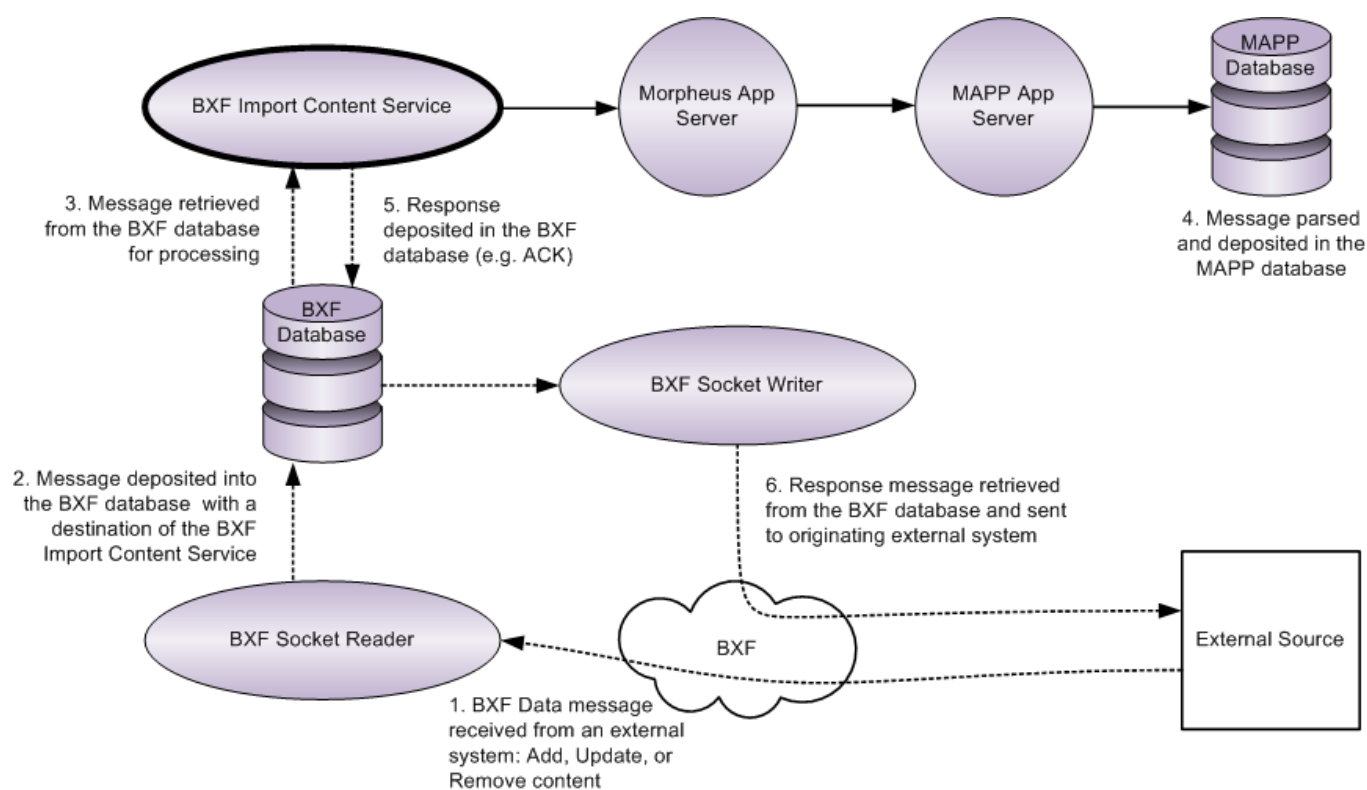


Fig. 2-22: BXF Import Content Service Logical Topology

## BXF Import Content Service Configuration Parameters

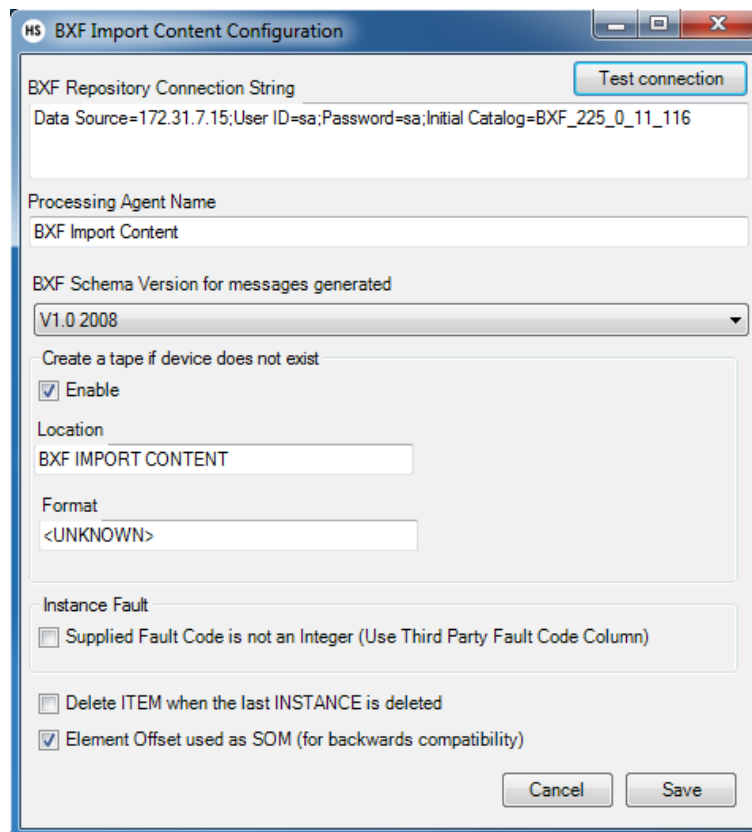


Fig. 2-23: BXF Import Content Service Configuration Window

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bx f catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

#### **- Processing Agent Name**

Enter a name that will identify the BXF Import Content Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Processing Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message to the external system, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages from the external system.

---

Note: It is recommended that the agent name is different to the instance name.

---

#### **- BXF Schema Version for Messages Generated**

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

#### **Create a Tape if Device Does Not Exist**

In the case of a message being received that specifies the physical location of content, a record for which does not exist in the MAPP database, then this function sets a default device type of 'Tape'. A tape record is created in the 'Storage Location' table of the MAPP database, the name of which is that received in the message.

#### **- Enable**

Tick the checkbox in order to enable this function.

The default is disabled.

#### **- Location**

Enter a text string that specifies an arbitrary location for the tape - this will be entered against each record of this nature that is processed by the service.

#### **- Format**

Enter a text string that specifies a default tape format (e.g. Betamax, VHS).

### Instance Fault

A received message may also contain descriptions of faults relating to the content. The MAPP database uses a predetermined fault code numbering system - should a message be received with a fault description that is associated to an unrecognised fault code, then this function will create a new entry in the database.

#### **- Supplied Fault Code is not an Integer**

Default: disabled

MAPP fault codes are uniquely integers. Should fault codes from a third party not match those in the MAPP database, then a prior association can be made in order for them to be registered in the MAPP Fault Code Table.

When this function is enabled, third party fault codes are registered in the MAPP database matched to a standard fault code, as shown below.

*Table: 2-1*

FAULT_CODE	FAULT_DESC	THIRD_PARTY_FAULT_CODE
MAPP Code	Common Fault Description	Received External Fault Code

---

Note: Each third party fault code must be associated to a MAPP fault code in the MAPP database during system commissioning.

---

#### **- Delete ITEM When the Last INSTANCE is Deleted**

Default: disabled.

When enabled, this function will delete the entry in the 'Item' table once all instances of the content have been deleted from the 'Instance' table.

---

Note: Within the MAPP database, an 'Instance' table refers to both an 'Item' table and a 'Storage Location' table:

---

- Original content has a single entry in the 'Item' table
  - Instances of the content in the 'Item' table have entries in the 'Instance' table
  - The instance is on a device for which there is an entry in the 'Storage Location' table
-

**- Element Offset Used as SOM (for backwards compatibility)**

This feature was added in Morpheus v5.9.2

Default: disabled (un-ticked)

For soft segments, the original iteration of the Morpheus software to support the content `<offset>` element in a BXF message differed from the specification of the standard, in that it was interpreted as the Start Of Material (SOM). The `<offset>` element in the BXF message would therefore have to contain a SOM for each soft segment, and not the offset from the main clip as would be expected.

The software was modified in v5.2.3 in order to reflect the correct interpretation of the BXF standard, i.e. the `<offset>` element now defines the start point of each soft segment relative to the beginning of the main clip.

This feature provides the ability to switch between both modes. When enabled, the original interpretation of the `<offset>` element is supported in order to allow Morpheus systems to be migrated from pre-v5.2.3 software to v5.9.2 onwards without the need for customers to also have to modify their BXF traffic systems concurrently. Once the BXF traffic systems have been amended in order to support the correct interpretation of the `<offset>` element, then this function can be disabled.



## BXF Lock Service

When enabled, this service protects main events (and their sub-events) that are close to going on air against schedule updates contained in received BXF messages. A lock is applied to a main event when it falls within a predefined window of time, and according to particular operational conditions. The service is also responsible for removing the lock should the main event be moved or pushed outside of the lock window.

The service is enabled on a per channel basis.

---

Note: Despite the name of this service, it has no association with the BXF protocol. Its role is to set the event parameter 'BxfLocked' to a true or false state.

It is the responsibility of the 'BXF Schedule Service' to evaluate the presence of BXF locks on main events and determine whether or not changes should be applied from a received BXF message. If the 'BxfLocked' field is set to TRUE on a main event then it will not be updated - if the 'BxfLocked' field is set to FALSE, then the main event will be changed accordingly.

Concerning the BXF Schedule Service in this respect:

- The response to a received message containing only changes to locked events will be a NAK, the schedule will not be modified, and a warning is posted in the validation column of the editor against the locked events in order to inform the operator that the received updates were not applied.
  - The response to a message containing both changes to locked and to non-locked events will be an ACK, but changes will only be applied to non-locked events. A warning is posted in the validation column of the editor against the locked events in order to inform the operator that the received updates were not applied.
- 

With the 'BXF Lock' field set to True, the following icon is displayed in the 'Validation' column in the Editor against the locked main event (and its sub-events) in order to show that it is protected from BXF updates:



Fig. 2-24: BXF Locked Icon

## BXF Lock Service: Principle of Operation

The functionality of the BXF Lock service is shown in Figure 2-25 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

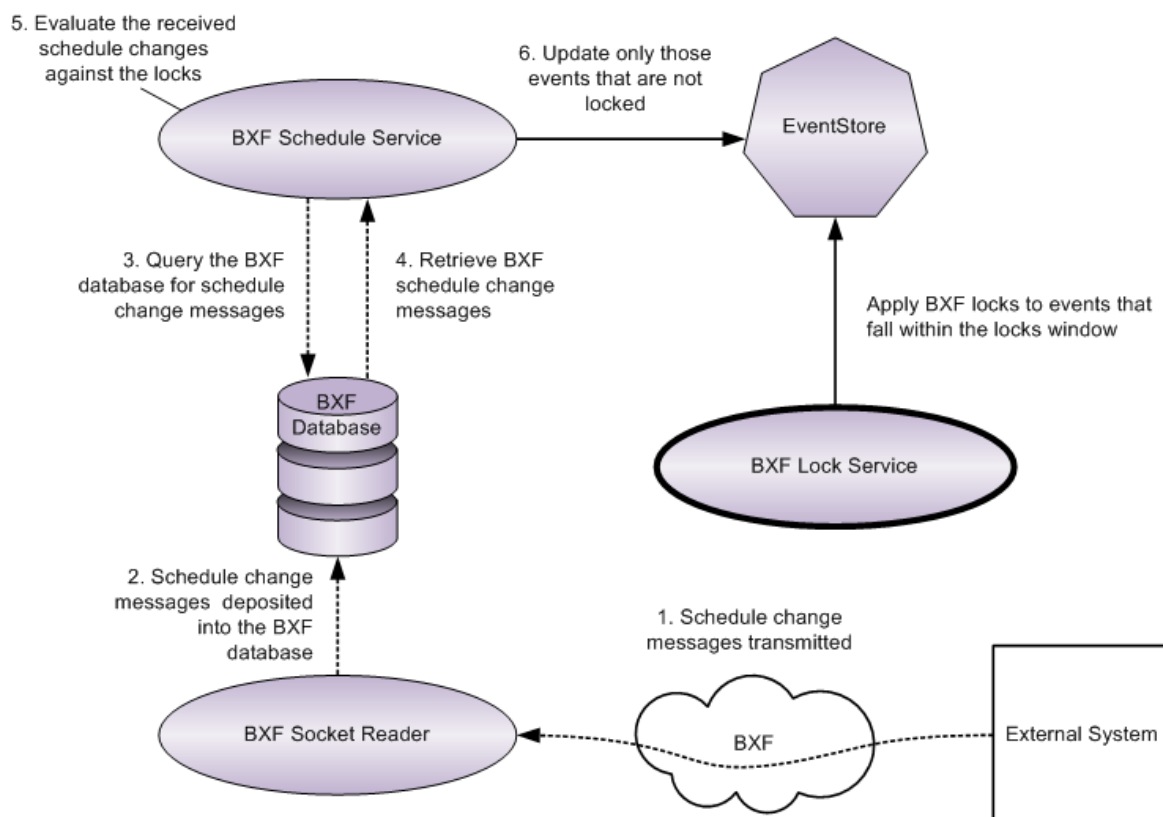


Fig. 2-25: BXF Lock Service Logical Topology

## Configuring the BXF Lock Service

The configuration requirement is in two parts:

- Configure the 'BxfLocked' event parameter
  - Create and configure an instance of the BXF Lock Service
1. Configure the 'BxfLocked' parameter on all main events for which the lock mechanism should apply - this parameter can be configured for each individual main event, or applied globally.

### Individually:

- a) In the Configurator, click on the **Configure Events** button.
- b) Select a main event and click on the 'All Parameters' tab, then click on the **Add Parameter** button.
- c) Configure the following properties, then click on **OK**:  
**Parameter Type:** *BooleanParamDef*  
**Parameter Name:** *BxfLocked*

---

Note: The parameter name is case sensitive.

---

### Globally:

- a) In the Configurator, click on the **Import/Export** button.
- b) In the 'ImportExport' window, click on the **Run Import Extensions** button.
- c) In the System Import Extensions window, click on the BXF tab, then click on the **Add BxfLocked Parameter to Every MainEvent and BreakHeaderEvent** button.

2. Open the 'Configure BXF Lock Service' window (Figure 2-26) by double clicking on the **BXFLockService** instance in the 'ShellHost Configuration' window (alternatively, highlight the instance of the service and click on the **Detailed Config** button). If an instance of the service does not already exist then refer to 'Configuration of HostShell Services in the Morpheus Engineer's Manual.

Parameter descriptions are provided after the configuration instructions below.

- a) Select the required BXF Lock Algorithm from the drop-down menu.  
If the 'SingleBxfLockWindowAlgorithm' has been selected, optionally tick the **Continue Locking Events In Consecutive Break Header** checkbox (this function is greyed out for all other lock algorithms).
- b) Highlight the channel to which the lock is to be applied from the list of those available (the name of the selected channel will appear under the 'Channel' heading at the bottom of the configuration window).
- c) Tick the **Enable Lock Windows** checkbox in order to activate the service for the selected channel.
- d) Configure the length of the **Live Lock Window** using the following format:  
<hours>:<minutes>:<seconds>:<frames>
- e) If the **LiveAndNonLiveLockWindowsAlgorithm** has been selected, configure the length of the **Non-Live Lock Window** (this parameter is not available if any of the other BXF Lock Algorithms are selected).

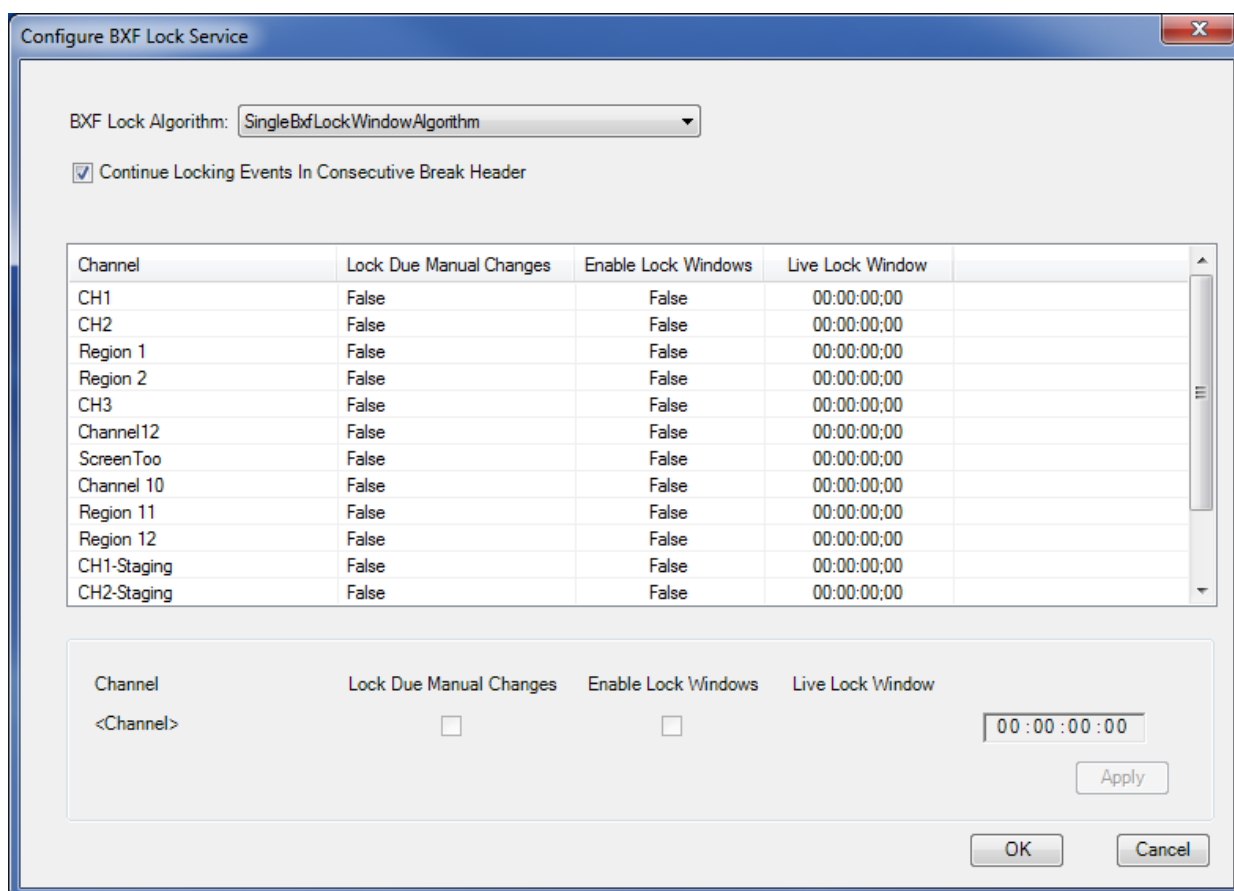


Fig. 2-26: BXF Lock Service Configuration Window

One of four types of operational condition, known as 'BXF Lock Algorithms', can be selected from a drop-down menu. The configuration and mode of operation of each is described below.

#### - LiveAndNonLiveBxfLockWindowsAlgorithm

Refer to Figure 2-27.

- Lock any schedule that is 'Live' (any of the main events are marked as being LIVE) if the start time to air of the first event falls within the **Live Lock Window**
- Lock any schedule that is not 'Live', if the start time to air of the first event falls within the **Non-Live Lock Window**

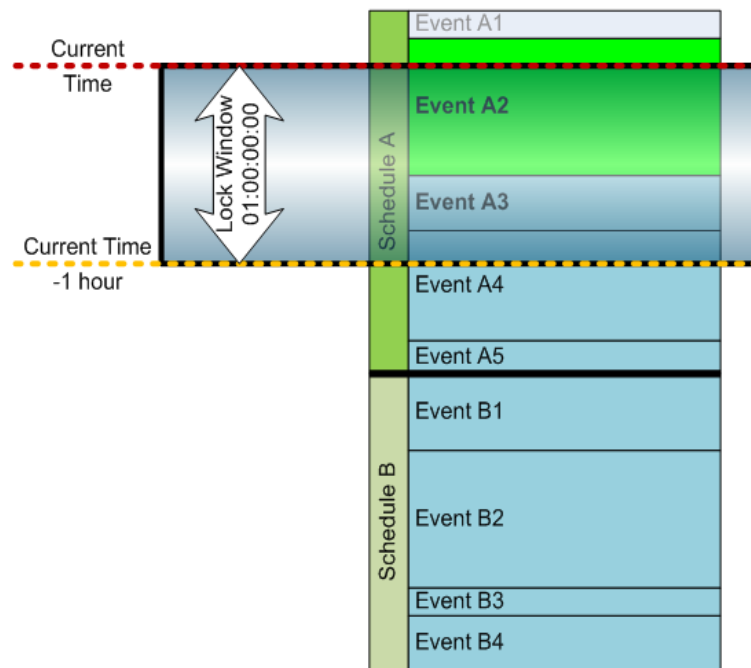


Fig. 2-27: BXF Lock Service - Live & Non-Live Lock Window Example

### - SingleBxfLockWindowAlgorithm

Refer to Figure 2-28.

If the on-air (PGM) event is not in a break header, then the service assumes that the preset (PST) event is in a Break Header. When this is the case, this algorithm sets the starting point of the **Lock Window** at the beginning of the first main event in the break header, and it will lock as many main events within that break header as fall within the **Lock Window**.

- **Continue Locking Events In Consecutive Break Header**

Note: Only available when the following BXF Lock Algorithm is selected:

**SingleBxfLockWindowAlgorithm**

Checkbox. If the **Lock Window**, as calculated in the point above, extends beyond the last main event in the break header, and there are back-to-back break headers, then a new lock window will start at the first main event in the next break header, and it will lock as many events in that break header as fall within the **Lock Window**.

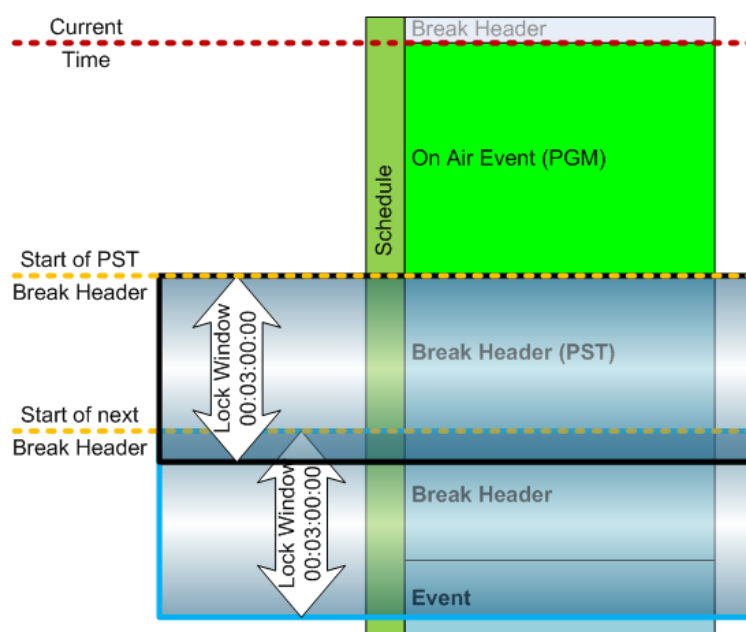


Fig. 2-28: BXF Lock Service - Single BXF Lock Window Algorithm  
(PGM event is not in a break header - example)

If the on-air (PGM) event is a main event within a break header, the **Lock Window** will be applied immediately, starting at the current time - all events with a start time that falls within the window are locked. If the **Lock Window** extends beyond the last main event in the break header, and there are back to back break-headers, and the **Continue Locking Events In Consecutive Break Header** checkbox is ticked, then a new **Lock Window** will start from first main event in the next break header, and it will lock as many main events within that break header as fall within the **Lock Window** (refer to Figure 2-29).

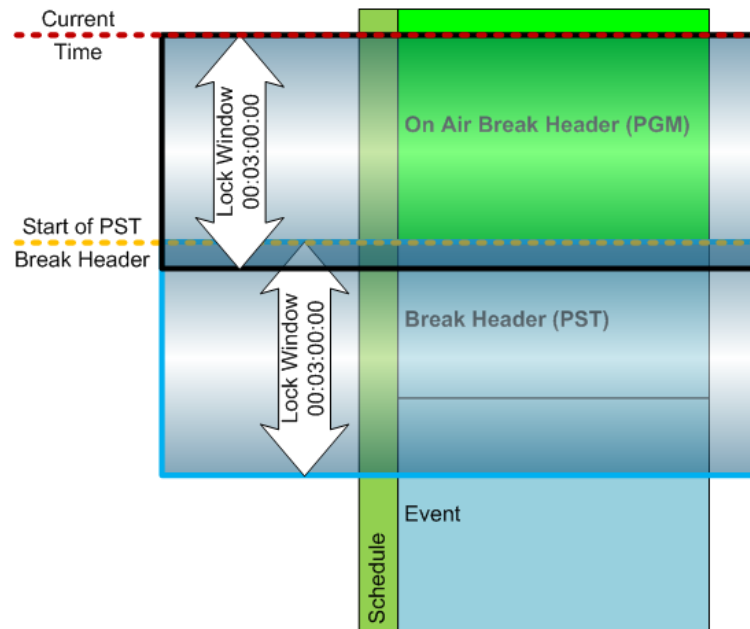


Fig. 2-29: BXF Lock Service - Single BXF Lock Window Algorithm  
(PGM event is in a break header - example)

#### - LockByTime

Locks top level events only, when their start time falls within the configured time in the **Lock Window**.

#### - LockCurrentDayAndByTime

Locks top level events only, for the rest of the current calendar day and for a configurable amount of time into the next day as specified by the time configured in the **Lock Window**.

#### - Lock Due Manual Changes (optional checkbox).

Default: disabled.

If enabled (ticked), the entire schedule is automatically locked if an operator makes a manual change to any event within it using the Editor - the schedule is locked at the point in time at which the change is made, even if this is ahead of the **Lock Window**.

Applied on a per channel basis.

**- Enable Lock Windows**

Enable the configured BXF lock, having set the duration of the 'Live Lock Window' (below).

Applied on a per channel basis.

**- Live Lock Window**

Applies only to events of type 'Live'.

Defines the point in time at which to apply the lock, ahead of the time to air of the event in the schedule that is applicable to the chosen algorithm. Use the following format:

<hours>:<minutes>:<seconds>:<frames>

Applied on a per channel basis.



## BXF Media Change Notifier Service

The purpose of the BXF Media Change Notifier service is to maintain synchronisation between the local MAPP database and the database on an external system.

The BXF Media Change Notifier service queries the 'Item Change' and 'Instance Change' tables in the MAPP database, in order to retrieve any alterations to material (i.e. the actions attributes Add, Update, and Remove). It optionally filters the results before generating BXF content change messages that are deposited into the BXF database for posting to an external system.

---

Note: A maximum of 100 content change entries are retrieved by this service following an individual query - each change is transmitted to a destination as an individual message.

---

---

Note: The BXF Media Change Notifier Service is only responsible for retrieving messages from a configured location and depositing them in the BXF database for collection by a service such as a BXF Socket Writer service.

---

The service queries the MAPP database and generates BXF content change messages without solicitation.

---

### BXF Media Change Notifier Messages for Program Content and Non-Program Content

There are subtle, but important, differences between the structure of a BXF message for Program Content and for Non-Program content.

The following examples illustrate the different message structures:

- **Program Content**

Content that is typically long in duration and excludes non-program types such as commercials.

```
<Content>

  <ProgramContent>

    <ContentMetadata>

      !--Describes the unique properties of a single clip or the
      shared properties of soft segments

    </ContentMetadata>

    <Elements>

      <Element>

        !--Describes a soft segment if it exists (optional)

      </Element>

      <Element>

        !--Describes another soft segment if it exists
        (optional)

      </Element>

    </Elements>

  </ProgramContent>

</Content>
```

The <ContentMetadata> node contains information relating to the overall content (single clip or soft segments) such as <HouseNumber>, <Name> and the action attribute that describes the change required on the content (i.e. 'add', 'update', or 'remove'); it is used in exactly the same manner as for non-program content.

---

Note: If the program content is a single clip then the <Elements> node is not required.

---

- **Non Program Content**

Short form content, such as a commercial, that is interspersed between the program content.

```
<Content>
  <NonProgramContent>
    <Details>
      <SpotType>User_defined_string</SpotType>
    </Details>
    <ContentMetaData>
      !--Describes the unique properties of the <SpotType>
      element
    </ContentMetaData>
  </NonProgramContent>
</Content>
```

The `<ContentMetadata>` node contains information relating to the `<SpotType>` element, such as `<HouseNumber>`, `<Name>` and the action attribute that describes the change required on the content (i.e. 'add', 'update', or 'remove'); it is used in exactly the same manner as for program content.

The `<SpotType>` node describes an interstitial, such as a commercial; it can be any user-defined string, as long as it matches with that of the destination external system.

---

Note: The `<EmbeddedNonProgramContent>` node is not supported.

---

## BXF Media Change Notifier Service: Principle of Operation

The functionality of the BXF Media Change Notifier service is shown in Figure 2-30 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

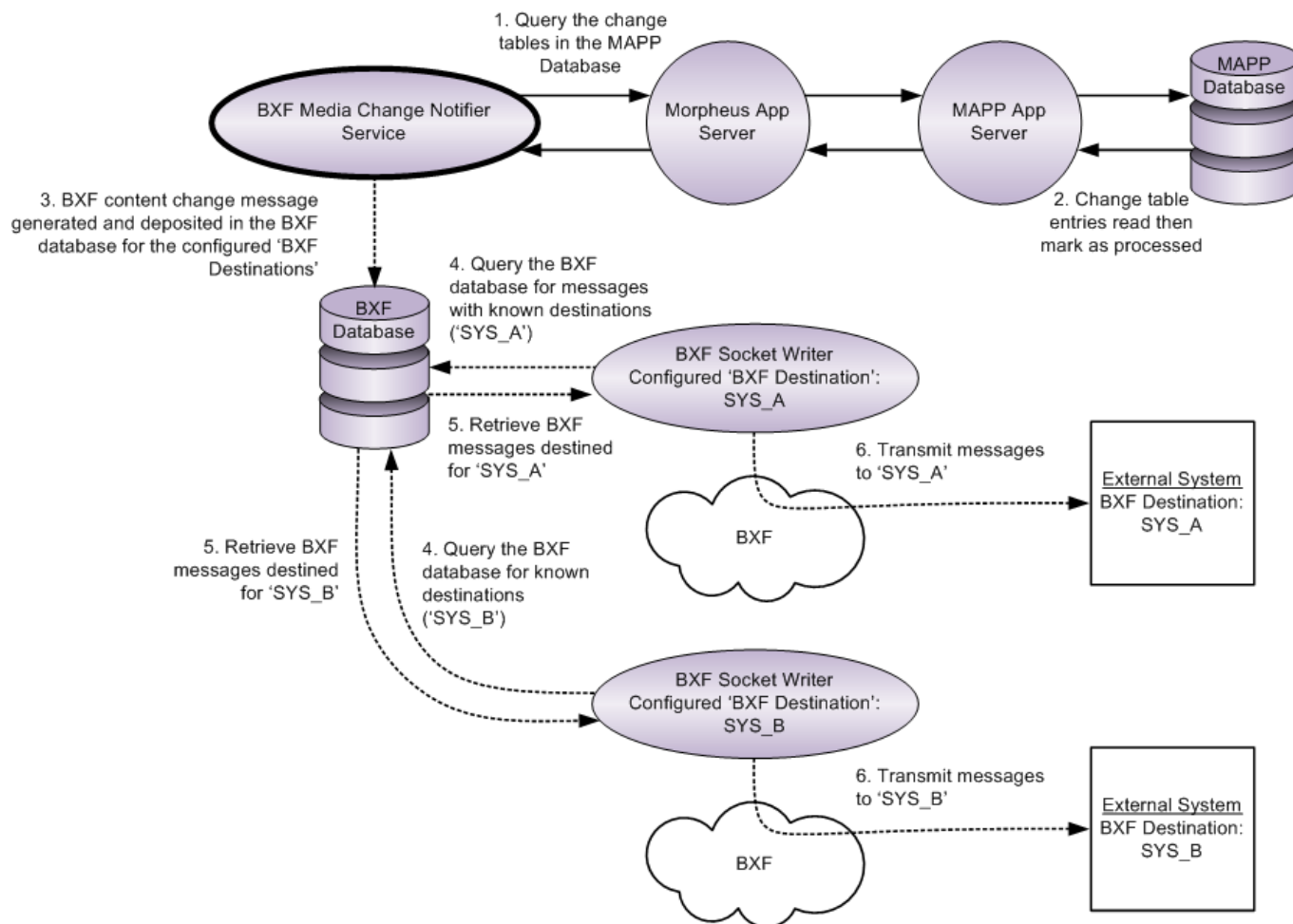


Fig. 2-30: BXF Media Change Notifier Service Logical Topology

This service generates a 'BXF Content Update' message.

Note: Despite two BXF Socket Writers being shown in Figure 2-30, a single instance of that service is capable of writing to multiple destinations. The number of BXF Socket Writers required will depend upon system design criteria.

## BXF Media Change Notifier Service General Configuration Parameters

Fig. 2-31: BXF Media Change Notifier Service Configuration Window

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxr catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

Note: The properties for this string will have been configured at the time of the database creation.

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Processing Agent Name

Enter a name that will identify the BXF Media Change Notifier Service as a processing agent of BXF media change messages - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Processing Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message to the external system, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages from the external system.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - Generate Message by Master Material ID

Default: disabled (un-ticked)

The MAPP database treats any Material ID as a single clip, therefore main material and soft segments are individually identified. This feature allows the service to generate a BXF message in one of the two following manners:

- Disabled: only the Material IDs of those clips that have changed will be entered into the message
- Enabled: the service scans the MAPP database for any clip that has the same Long\_File\_Id as the one that has changed, then generates a message containing the main Material ID and the IDs of any soft segments that relate to it

### - Delay on Message Creation (Debounce)

Default: 10 seconds.

Recommended setting: 10 seconds.

The interval in seconds between the creation of media change messages by this service. The configured delay starts at the point at which the previous message creation process has finished.

Regardless of the value attributed to this function, the time required for the transmission of a message from source to destination is approximately 5 seconds.

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### - Device Path

From the drop-down menu, select the format of the path that will be used in a generated message in order to define the device on which the changed content is stored. The format indicates the MAPP database 'Server Type' field from which the values will be extracted in order to define the path. Select from the following options:

- **StorageLocation**: concatenate the value in the DEVICE\_NO field with that of the STORAGE\_LOCATION field in order to create the following path format:  
\\DEVICE\_NO/STORAGE\_LOCATION
- **StorageLocationUNC**: use the value in the STORAGE\_LOCATION\_UNC field only, in order to create the following path format:  
\\STORAGE\_LOCATION\_UNC

### General Tab: BXF Destinations

Enter the names of the BXF destinations to which to send BXF media change messages. For every change message, one copy will be sent to each of the configured destinations.

- **Add**

- a) Enter the name of the BXF destination to which messages are to be sent.

---

Note: Parameters are case sensitive.

---

- b) Click on **Add**. The destination is displayed in the 'BXF Destinations' pane.

- **Replace**

Highlight the destination entry to be changed - the text entry box will be populated with the entry. Enter the new destination and click on **Replace**.

- **Delete**

Highlight the destination entry to delete then click on **Delete**.

## BXF Media Change Notifier Service Filters

The filtering mechanism provides a method of selecting the detail that will be reported in a message sent from this service to an external system. Filtering is based upon the existence of a material type on specific devices, and the change that was performed upon it. Refer to the flowchart in Figure 2-32 for an understanding of the filter principles.

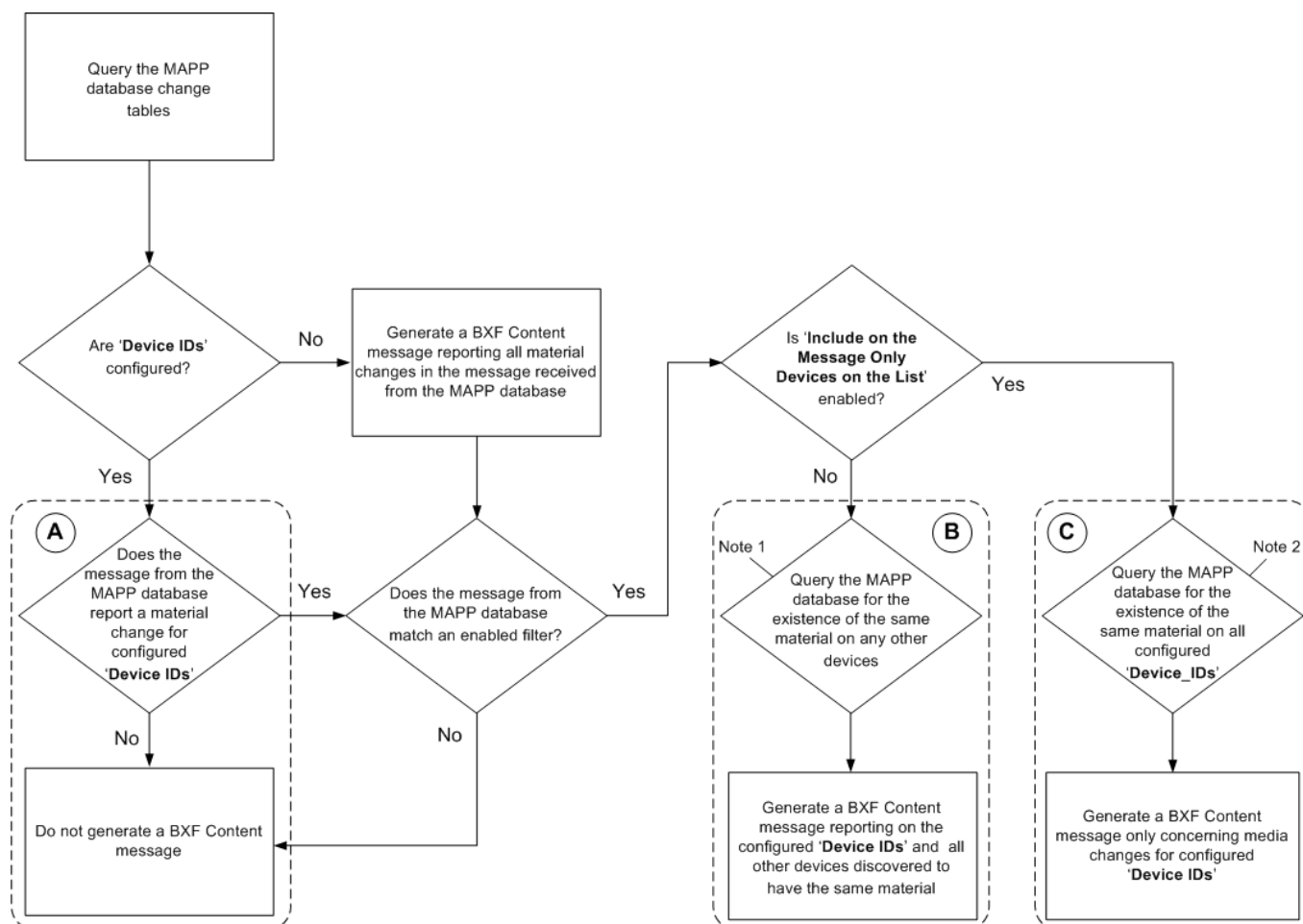


Fig. 2-32: BXF Media Change Notifier Configuration: Filtering Principles

### Notes on the Flowchart

**Note 1:** the service queries the MAPP database for a second time in order to ensure that the BXF message that will be generated informs the destination external service of the existence of all instances of the material, on all devices, whether they have been changed or not.

**Note 2:** in the case where changes were only reported for a subset of the configured Device Id's, the service queries the MAPP database for a second time in order to establish the status of the material on all configured Device Id's.

**Flow A:** do not generate a BXF Content message unless the message from the MAPP database reports a material change for configure Device Id's, otherwise proceed to Flow B or Flow C.

**Flow B:** if 'Include on the Message Only Devices on the List' is disabled then generate a BXF message for the existence of the same Material Id on all devices in the MAPP database.



**Flow C:** if 'Include on the Message Only Devices on the List' is enabled then generate a BXF message for the existence of the same Material ID only for the configured 'Device Id's.

Access the filter configuration page by clicking on the 'Filter' tab, as shown in Figure 2-33.

Fig. 2-33: BXF Media Change Notifier Configuration: Filters

Note: Only a single filter combination (Material Types / Operation Types / Instance) can be created, and this will apply to all devices in the 'Device List'. It is not possible to create individual filters for different Device Ids.

### - Add / Replace / Delete

Add, replace, or delete the IDs of devices that this functionality will filter on when generating the BXF message from change records retrieved from the MAPP database.

Should a query of the MAPP database tables indicate a change to material on one of the added Device Id's, then a second query is conducted in order to discover if any other known devices have a record of the same material in the MAPP database (whether it has been changed or not).

---

Note: This behaviour is altered if **Include on the Message Only Devices on the List** (below).

---

- **Add**

a) Enter a Device Id in the text box.

---

Note: Parameters are case sensitive.

---

b) Click on **Add**. The Device Id is displayed in the 'Device List'.

---

Note: Should no devices be specified then this service will generate a BXF message based upon all of the material changes discovered in the MAPP database, regardless of the device.

---

- **Replace**

To modify an existing Device Id, click to select it and it will appear in the text box. Alter the Id then click on the **Replace** button.

To remove a Device Id, click to select it and click on the **Delete** button.

### - Include on the Message Only the Devices on the List

Default: disabled (un-ticked).

When enabled, the standard behaviour that is consistent with the Device Id list is altered in the following manner: when the MAPP database is queried for the second time, only those material change records that apply to the devices in the Device Id list will be included in the BXF message.

### - Material Type

For all of the Device Id's that have been added, select the Material Types against which to perform a match. Select from the following (by default, all are selected):

- JUNCTION
- PROGRAMME
- COMMERCIAL
- LIVE
- LIVERECORD

### - Operation Type

Refine the information that will be reported based upon the actions that have been applied to the material on the Device Id's that have been configured. Select from the following available actions (by default, all are selected):

- INSERT
- UPDATE
- DELETE

### - Filter on Instance Column

With no other filters configured, this function will operate independently; alternatively it will operate in conjunction with others should they have been configured.

From the drop-down menu, select the name of a column from the 'Instance' table of the MAPP database upon which to filter. Once selected, set a value type from those available using the radio buttons (below).

---

Note: Only an instance table column name can be selected.

---



---

Note: When configured, this service will only generate a BXF message when changes occur on the specified 'Instance' table column.

---

The default is **[NONE]**.

#### Value type radio buttons:

- **Null:** include an entry in the BXF message for the specified 'Instance' table column name if its value is null  
Default.
- **Not Null:** include an entry in the BXF message for the specified 'Instance' table column name if it has a value other than 'Null'
- **Value:** include an entry in the BXF message for the specified 'Instance' table column name if it has the configured alphanumeric value only.

## **BXF Media Metadata Import Service**

---

Note: This service has been deprecated.

---

## **BXF Multiplex Message Service**

A BXF Query message distribution and aggregation service between 'core and remote' MAPP systems. The BXF Multiplex Message Service is designed to duplicate a BXF Query message for material that has been received from an external source into a core BXF system, for transmission to individual remote systems for processing. The responses from the remote systems are subsequently aggregated by the BXF Multiplex Message Service and returned to the external source as a single message.

## BXF Multiplex Message Service: Principle of Operation

The functionality of the BXF Multiplex Message service is shown in Figure 2-34 in relation to the other services upon which it relies in order to fulfill its role. The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

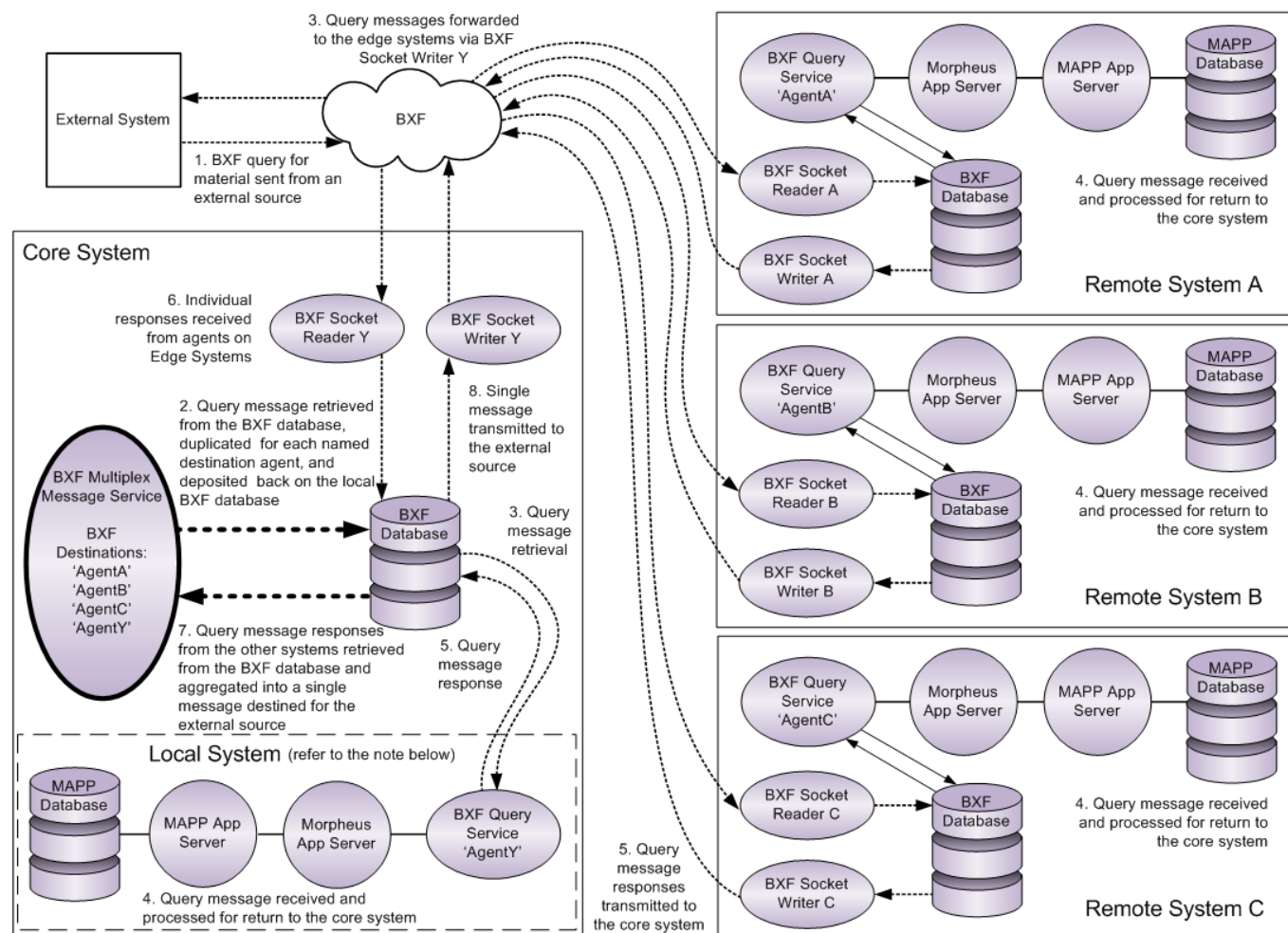


Fig. 2-34: BXF Multiplex Message Service Logical Topology

Note: Also shown, in the inset in Figure 2-34, there is a Local System adjacent to the Core that is also addressed by the BXF Multiplex Message service. As it is local, there is no requirement for its BXF Query service to communicate through a BXF Socket Writer.

Note: The infrastructure may be designed in order for the core services, the remote services, and the external system to communicate over the same or separate BXF Socket Writers and Readers. The infrastructure shown in Figure 2-34 is designed to use the same Writers and Readers, depicting a typical deployment.

## BXF Multiplex Message Service Configuration Parameters

Fig. 2-35: BXF Multiplex Message Configuration

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxm catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Processing Agent Name

Enter a name that will identify the BXF Multiplex Message Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Processing Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message to the external system, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages from the external system.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - Time Out (for receive reply)

Default: 0

Recommended: 10 seconds.

The time limit that is imposed on the remote BXF Query services within which to respond. Only those responses received within the configured time will be included in the multiplex message returned to the external system by the core system.

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### - Add / Replace / Delete BXF Destinations

The names of the BXF Query Service agents that will receive a duplicated message from this service - no message will be sent to any agent that is not listed.

- **Add**

- a) In the text box, enter the name of the destination service to which messages are to be sent.
- b) Click on **Add** - the destination service is added to the 'BXF Destinations' pane.

- **Replace**

Highlight the destination entry to be changed - the text entry box will be populated with the entry. Enter the new destination and click on **Replace**.

- **Delete**

Highlight the destination entry to delete then click on **Delete**.



## BXF Query Service

A service that processes BXF Query Request messages, addressed to its agent name from another system (internal or external), and interrogates either the MAPP database or the Morpheus EventStore for the required information (a valid received message can only contain either `<content>` elements, in which case the service will interrogate the MAPP database, or `<schedule>` elements, in which case the service will interrogate the EventStore). The results are converted into a BXF Query Response for transmission to the originating system. As a passive service, it responds only to received requests.

---

Note: As the BXF Query service is addressed by its Agent Name, that must be configured on the external system.

---

---

Note: In a Multi-Site Sync or an N+1 environment, the BXF Query service is only used for the initial schedule synchronisation procedure, or for a re-synchronisation.

---

## BXF Query Service: Principle of Operation

The functionality of the BXF Query service is shown in Figure 2-36 in relation to the other services upon which it relies in order to fulfill its role. The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

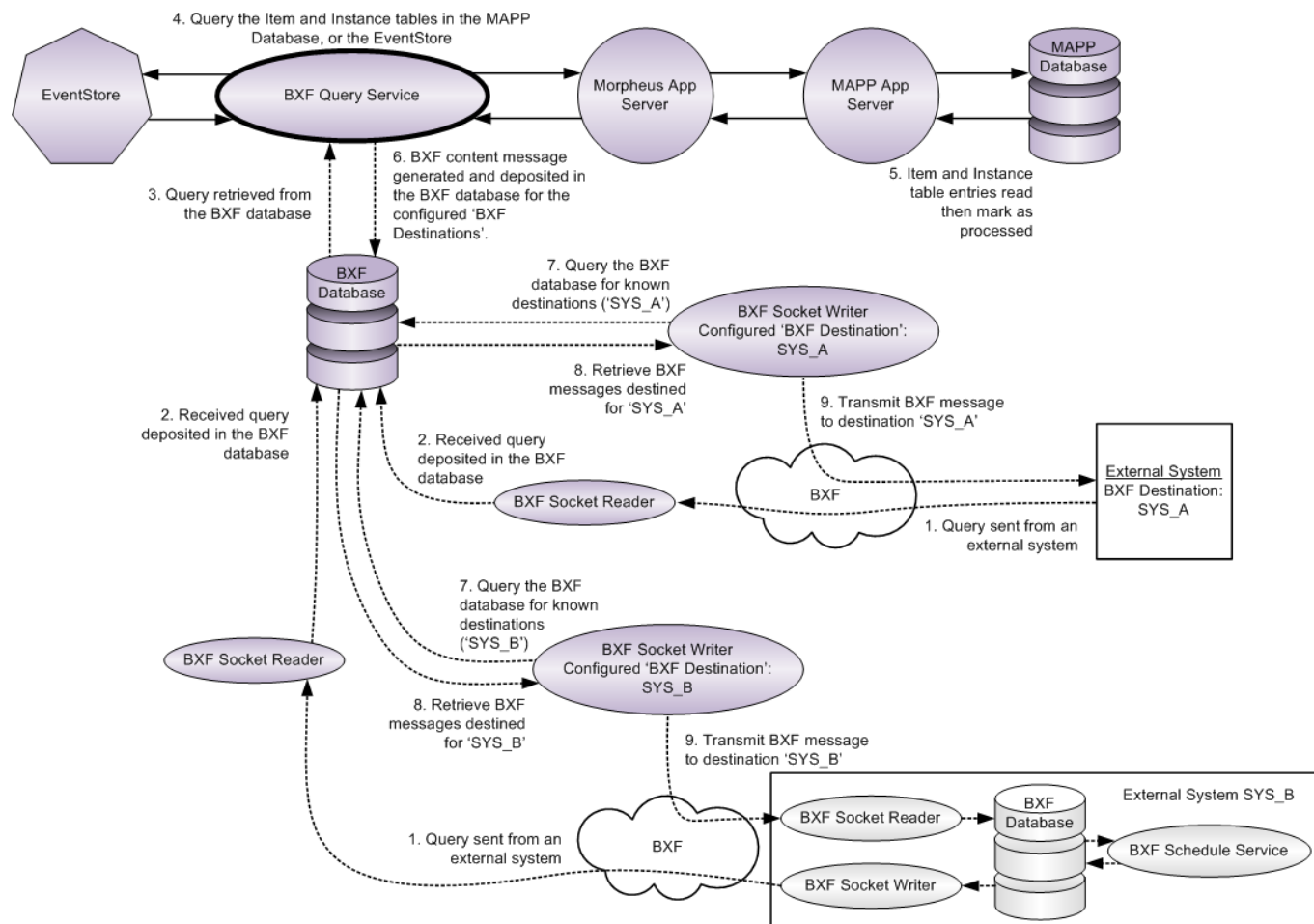


Fig. 2-36: BXF Query Service Logical Topology

Inside the BXF message, a query has the following element syntax, expressed as paths (example):

```
<BxfQuery>
  <WhereClause>Schedule/Channel/Name="Channel 1"</WhereClause>
  <ReturnStructure>Schedule/*</ReturnStructure>
</BxfQuery>
```

The **<WhereClause>** element contains identifiers that specify the nature of the data required from the MAPP database or the EventStore.

The **<ReturnStructure>** element contains identifiers that specify the BXF format in which the result of the query must be returned to the requesting system.

Note: The query type can be determined from the identifier in the **<ReturnStructure>** element.

## Query Types

There are four types of query, three of which are processed by this service, as follows:

---

Note: The Content Transfer query type is processed by the MAPP BXF Transfer Request Service (refer to Content Transfer Requests in the section on the MAPP BXF Transfer Request Service).

---

### 1. **BXF Query Content** (querying content)

Returns content information that is requested according to the entries of the <WhereClause> element in the received message.

The following identifying strings are supported for a query of type 'Content' - they are shown against the MAPP database fields to which this service will translate them (Table 2-2):

Table: 2-2

BXF Identifying String	MAPP Field
House Number	MaterialID
ReferenceName	DeviceID
ReferenceName AND PathName	DeviceID AND LongFileID
ReferenceName AND RecordedDate* AND Encoded	DeviceID AND RecordedDate AND Encoded
Content**	All 'Items' and 'Instances' thereof

---

Note: \*RecordedDate and Encoded are not BXF identifying strings, they are MAPP fields that are consequently inserted into the 'PrivateInformation' element of the BXF Query message (refer to the example shown below).

---

---

Note: **\*\*The BXF standard recommends that messages be limited to a maximum size of 2MBytes. An *All Items and Instances* 'Content' query response, by its nature, can be in excess of the imposed limit, in which case the BXF service must perform the following actions:**

- Fragment the message
- Set the 'chunkingFlag' in the BXF message to 'True' in order to indicate that the message is being transmitted in multiple parts:

`chunkingFlag = true`

- Indicate the number of each message relative to the others (e.g. 2 of 10):

`itemNumber = 'positive integer' (the number of the message)`

`totalItems = 'positive integer' (the total number of messages)`

This BXF recommendation has only been applied to the BXF Content Query Reply message of this service.

All Host Shell BXF services will accept messages greater than 2MBytes in size, and it is reasonable to expect that external BXF services will do the same.

The 2MBytes limit is configured by setting the recommended value for 'Number of Material to Break the Message' on the 'Content (Item / Instance)' tab of this service.

---

An example of each supported query of type 'Content' are shown over the page.

### BXF Query Content Examples

HouseNumber (highlighted in red)

```
<BxfQuery>  
<WhereClause>Content/ProgramContent/ContentMetaData/ContentId/HouseNumber="000000000020"  
</WhereClause>  
<ReturnStructure>Content/*</ReturnStructure>
```

ReferenceName (highlighted in red)

```
<BxfQuery>  
<WhereClause>Content/ProgramContent/ContentMetaData/Media/MediaLocation/Location/  
AssetServer/ReferenceName="INGEST_SVR"</WhereClause>  
<ReturnStructure>Content/*</ReturnStructure>
```

ReferenceName AND Pathname (highlighted in red)

```
<BxfQuery>  
<WhereClause>Content/ProgramContent/ContentMetaData/Media/MediaLocation/Location/  
AssetServer/ReferenceName="INGEST_SVR" and  
Content/ProgramContent/ContentMetaData/Media/MediaLocation/Location/AssetServer/  
PathName="00002828"</WhereClause>  
<ReturnStructure>Content/*</ReturnStructure>
```

ReferenceName AND RecordedDate AND Encoded (highlighted in red)

---

Note: RecordedDate and Encoded are MAPP fields not supported in the BXF standard, and they are therefore inserted into the 'PrivateInformation' element of the BXF Query message (as shown highlighted in blue in the example below).

---

```
<BxfQuery>  
<WhereClause>Content/ProgramContent/ContentMetaData/Media/MediaLocation/Location/AssetServer/  
ReferenceName="OMNEON3" AND#  
Content/ProgramContent/ContentMetaData/Media/MediaLocation/PrivateInformation/Snell/  
IntanceInfo/RecordedDate>="2006-12-11 00:00:00.0Z" AND  
Content/ProgramContent/ContentMetaData/Media/MediaLocation/PrivateInformation/Snell/  
IntanceInfo/RecordedDate<="2006-12-13 00:00:00.0Z" AND  
Content/ProgramContent/ContentMetaData/Media/MediaLocation/PrivateInformation/Snell/  
QualityControl/Encoded="1"</WhereClause>  
<ReturnStructure>Content/*</ReturnStructure>
```

All Content (highlighted in red)

```
<BxfQuery>  
<WhereClause>Content="*"</WhereClause>  
<ReturnStructure>Content/*</ReturnStructure>  
</BxfQuery>
```

## 2. **BXF Query Schedule** (querying schedules)

Returns schedule information that is requested according to the entries of the `<WhereClause>` element in the received message, and on a per channel basis.

The following identifying strings are supported for a query of type 'Schedule' - they are shown against the EventStore fields to which this service will translate them (Table 2-3):

Table: 2-3

BXF Identifying String	EventStore Field
Channel	Channel Name
Channel AND Period	Channel Name AND Actual Start Time
Channel AND ScheduleId	ChannelID AND *External ID
ScheduleId (list of ScheduleId's)	*External ID

Note: \* The 'External ID' (also known as the 'BXF ScheduleId') is extracted from the Schedule Information Object (SIO).

### **BXF Query Schedule Examples**

#### Channel (highlighted in red)

Return all schedules, including their events, for the channel specified in the received BXF message.

`<BxfQuery>`

`<WhereClause>Schedule/Channel/Name="BTS1"</WhereClause>`

`<ReturnStructure>Schedule/*</ReturnStructure>`

`</BxfQuery>`

#### Channel and Period (highlighted in red)

Return all schedules, including their events, for the channel and time period specified in the received BXF message.

`<BxfQuery>`

`<WhereClause>Schedule/Channel/Name="401" AND  
Schedule/@scheduleStart>="2014-12-20T15:30:00.0Z" AND  
Schedule/@scheduleEnd<="2014-12-20T16:00:00.0Z"</WhereClause>`

`<ReturnStructure>Schedule/*</ReturnStructure>`

`</BxfQuery>`

### Channel and Schedule ID (highlighted in red)

Return the schedule of events for the channel and ScheduleID specified in the received BXF message.

```
<BxfQuery>
  <WhereClause>Schedule/Channel/Name="401" AND
  Schedule/@scheduleId="urn:uuid:00000000-0000-0000-0000-
  0e3162504172"</WhereClause>

  <ReturnStructure>Schedule/*</ReturnStructure>
</BxfQuery>
```

### ScheduleID (list of ScheduleIDs, highlighted in red)

Return the schedules with the specified ScheduleIDs in the received BXF message.

```
<BxfQuery>
  <WhereClause>Schedule/@scheduleId="urn:uuid:00000000-0000-0000-0000-
  0e3162504172" OR Schedule/@scheduleId="urn:uuid:00000000-0000-0000-0000-
  0e3162504173"</WhereClause>

  <ReturnStructure>Schedule/Channel/*</ReturnStructure>
</BxfQuery>
```

---

Note: The information returned in the message from this service can be adapted using the <ReturnStructure> element, as follows:

- If the value of the <ReturnStructure> element is Schedule/\*, then the message will detail the schedule name and the events
- If the value of the <ReturnStructure> element is Schedule/Channel/\*, then the message will detail the schedule name only, excluding the events. This provides a method for an external system to query the existence of a particular schedule.

This element can be adapted in all of the BXF Query Schedule message type variations above.

---



### 3. **BXF Query AsRun** (querying AsRun records)

Returns AsRun information that is requested according to the entries of the `<WhereClause>` element in the received message and on a per channel basis over a specified period of time.

The following identifying strings are supported for a query of type 'AsRun' - they are shown against the EventStore fields to which this service will translate them (Table 2-4):

Table: 2-4

BXF Identifying String	EventStore Field
Channel AND Period	Channel Name AND Actual Start Time

#### **BXF Query AsRun Example**

Channel (highlighted in red)

Return the AsRun records for a nominated channel between the specified dates and times.

`<BxfQuery>`

```
<WhereClause>Schedule/Channel/Name="404" and
Schedule/@scheduleStart>="2011-07-12T14:40:00.0Z" and
Schedule/@scheduleEnd<="2011-07-12T15:59:59.0Z"</WhereClause>
```

```
<ReturnStructure>Schedule/AsRun/*</ReturnStructure>
```

`</BxfQuery>`

## Common Parameters

The following parameters are common to all tabs in the BXF Query Service Configuration window.

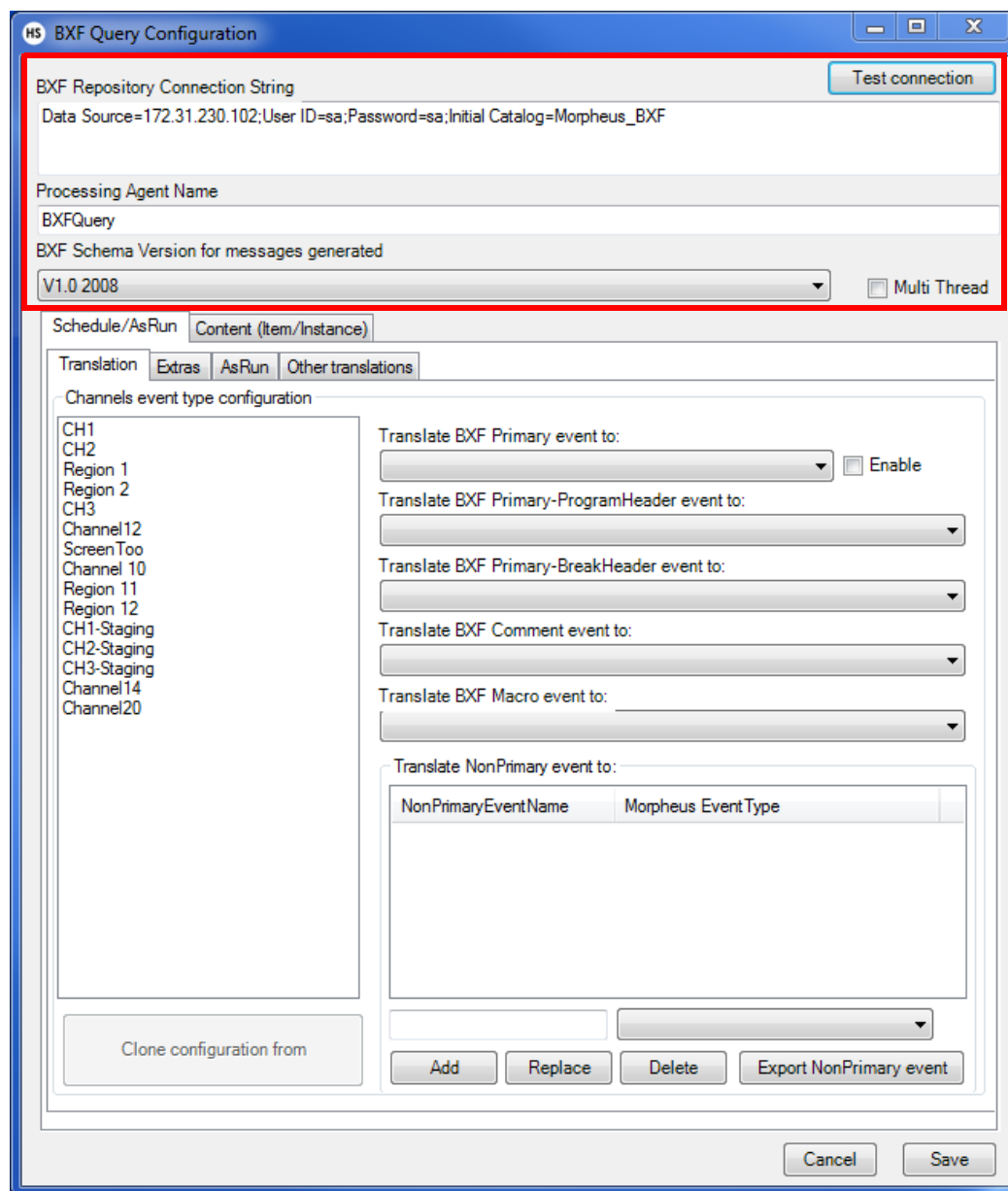


Fig. 2-37: BXF Query Service Configuration Window - Translation Tab

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxfl catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalog=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Processing Agent Name

Enter a name that will identify the BXF Query Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Processing Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message to the external system, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages from the external system.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### **- Multi Thread**

Default: disabled (un-ticked).

By default, when disabled, this service will process all query messages consecutively, in received order.

When enabled, this service will process query messages according to the following rules:

- Messages from different origins are processed concurrently
- Messages from the same origin are processed consecutively, in received order

---

Note: In order to maintain EventStore read / write performance, it is recommended that this feature is enabled for an instance of this service if it is configured to process only 'BXF Query Content' and 'BXF Query AsRun' messages from multiple origins, and limited to processing 'BXF Query Schedule' messages from only a single origin.

---

## Schedule / AsRun: Translation Tab

Channel specific parameters.

The BXF standard defines a number of default event types to which specific Morpheus event types must be matched in order for the service to build the BXF Query Response message with the requested information. The matching process is known as a 'Translation'.

Note: Configurations are applied on a per channel basis.

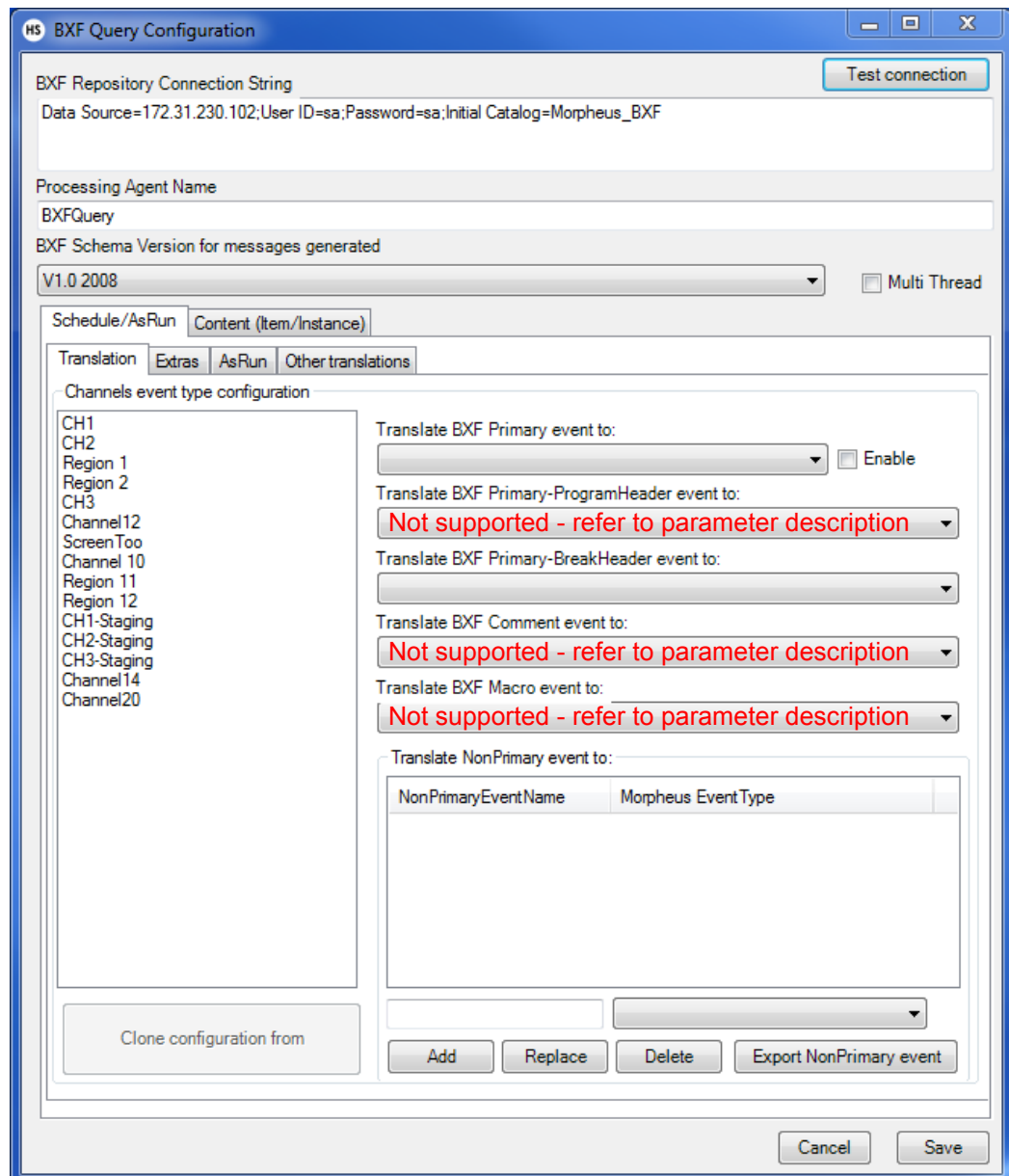


Fig. 2-38: BXF Query Service Configuration Window - Translation Tab

### - Enable

Default: disabled (un-ticked).

Enable the translation capability on a per channel basis.

If translations are not configured and enabled for a channel, any information for that channel that corresponds to a request in the message will be ignored. Under such circumstances, the service will behave in one of two ways:

- If a received query specifies the channel then the service will respond with a NAK.
- If the channel is not specified in the message, but information relating to it corresponds to the request, then such information will not be included in a response.

### - Channels Event Type Configuration

The list of Morpheus channels to which translations can be applied. This menu is populated automatically from the EventStore.

### - Clone Configuration From

Apply the set of existing primary and non-primary event translations on one channel to another:

1. Select the channel on which to apply the set of translations.
2. Click on the **Clone Configuration From** button.
3. The 'Select Source Channel' window is displayed (Figure 2-39).

From the drop-down menu, select the channel from which the translations will be cloned, then click on **Ok**.

The translation configuration is applied to the selected channel.

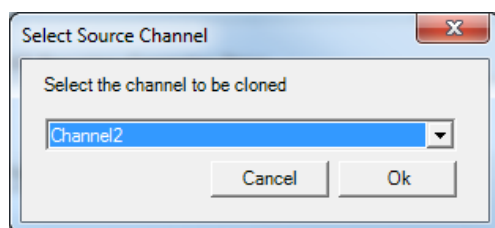


Fig. 2-39: Select Source Channel

---

Note: The receiving channel must support the event type related to the translation that is being cloned.

---

---

Note: This feature identifies whether or not the source event translations include the source channel name, and if so, that portion is changed to the name of the target channel during the cloning process.

---

**- Translate BXF Primary Event To**

No default.

Click on the channel(s) that this event type applies to, and from the drop-down menu select the Morpheus event type that will translate to a BXF Primary Event.

**- Translate BXF Primary-ProgramHeader Event To**

No default.

This BXF event type is not currently supported by the BXF Query Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

**- Translate BXF Primary-BreakHeader Event To**

No default.

Click on the channel(s) that this event type applies to, and from the drop-down menu select the Morpheus event type that will translate to a BXF Primary-BreakHeader Event.

**- Translate BXF Comment Event To**

No default.

This BXF event type is not currently supported by the BXF Query Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

**- Translate BXF Macro Event To**

No default.

This BXF event type is not currently supported by the BXF Query Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

### - Translate NonPrimary Event To

No defaults.

Non-primary event names (e.g. sub-events for GPIs and subtitles) are not specified in the BXF standard, and as such they are user definable.

---

Note: Any NonPrimaryEventName configured for this feature must match with that configured on the system from which messages will be received and responses transmitted.

---

For each channel that is configured with non-primary events add BXF translation entries as follows (Figure 2-40):

The screenshot shows a dialog box titled "Translate NonPrimary event to:". Inside, there is a table with two columns: "NonPrimaryEventName" and "Morpheus Event Type". The table is currently empty. Below the table, there are two input fields labeled "A" and "B". Field "A" is a text box, and field "B" is a drop-down menu. At the bottom of the dialog, there are four buttons: "Add", "Replace", "Delete", and "Export NonPrimary event".

Fig. 2-40: BXF Query Service - Translate NonPrimary Event To

- **Add**

- Click on a channel in order to select it.
- In the text box 'A', enter a name for the non-primary event that will be sent in a BXF message.

---

Note: Names are case sensitive.

---

- From the drop-down menu 'B', select the Morpheus event type that will translate to the non-primary event selected in box 'A'.
- Click on the **Add** button - the new translation entry is added to the table.

- **Replace**

To modify an existing translation entry, click on any part of it and the text box and drop-down menus will be populated with its current properties. Enter alternative properties then click on the **Replace** button.

- **Delete**

To remove a translation entry from the table, click on any part of it then click on the **Delete** button.



### - **Export NonPrimary Event**

Save, to a file, the non-primary event translations configured for a selected channel. This function is only provided for administrative records - there is no import capability to allow sharing of the configured translations.

1. From the channel list, select the one for which to export the translations.
2. Click on the **Export NonPrimary Event** button.
3. Windows Explorer is opened: enter a filename and destination folder, then click on the **Save** button.

---

Note: Only the non-primary events for the currently selected channel are exported.

---

## Schedule / AsRun: Extras Tab

If additional event parameters and their values, that relate to the schedule or AsRun, are not standard BXF elements and must therefore be included in the 'PrivateInformation' element of the BXF message, then these should be defined on the BXF Query Service **Extras** Tab, as shown in Figure 2-41.

Note: The 'PrivateInformation' element of the BXF message is supported in all BXF schemas.

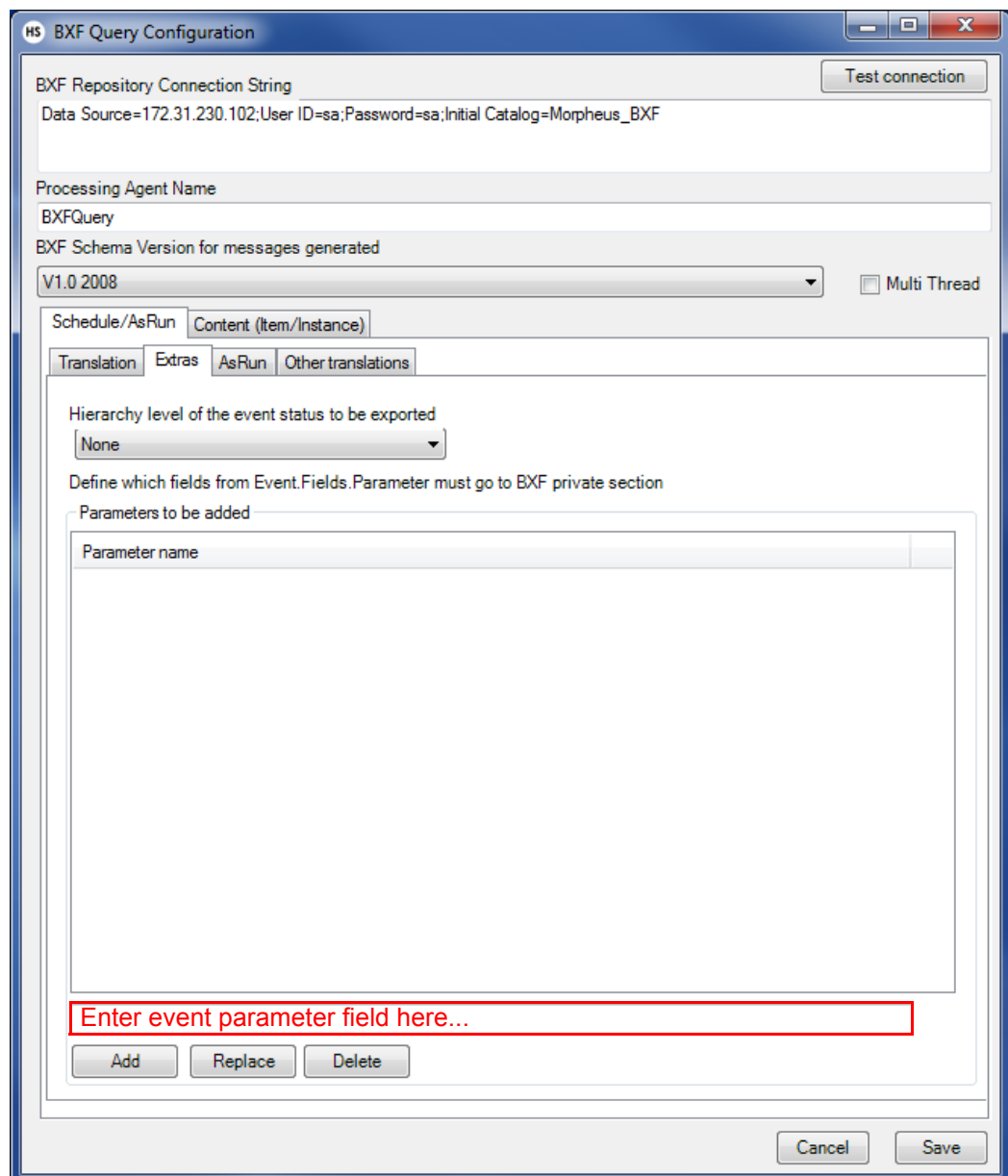


Fig. 2-41: BXF Query Service Configuration Window - Extras Tab

### - Hierarchy Level of the Event Status to be Exported

The severity level of event status log entries that are to be included in the BXF messages. Only those entries that have been generated at the severity level selected will be transmitted in a BXF message. Entries will be included in the 'PrivateInformation' element of the BXF message.

From the drop down menu select the required message severity level. The selected hierarchy level will apply to event status entries for all channels that are selected to use this service.

Select from the following:

- None
- Information
- Warning
- Error
- Critical

### - Parameters to be Added

- **Add**

- a) In the text box, enter the event parameter field to include in the BXF message, as defined in the event's **All Parameter Definitions** tab in the Configurator.
- b) Click on **Add** - the field name is added to the 'Parameters to be Added' pane.

---

Note: Parameter names are case sensitive.

---

- **Replace**

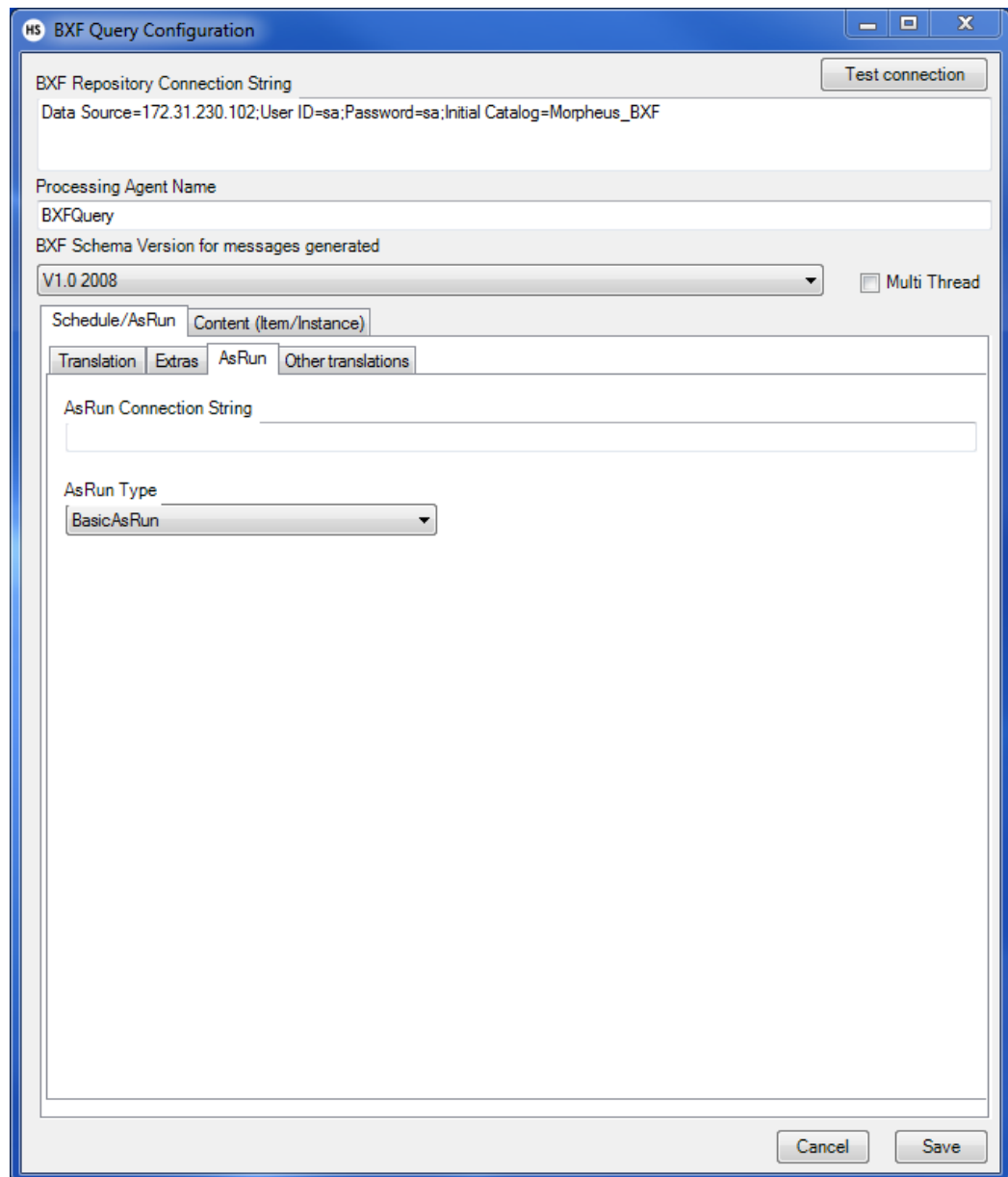
Highlight the event parameter field name entry to be changed - the text entry box will be populated with the entry. Enter the new event parameter field name and click on **Replace**.

- **Delete**

Highlight the event parameter field name to delete then click on **Delete**.

## Schedule / AsRun: AsRun Tab

The configuration parameters required to access the AsRun database, and to specify the level of AsRun information that will be sent in a BXF Query AsRun response message.



The image shows a Windows-style configuration window titled "BXF Query Configuration". It has a blue title bar with standard minimize, maximize, and close buttons. The window is divided into several sections. The top section is labeled "BXF Repository Connection String" and contains a text field with the value "Data Source=172.31.230.102;User ID=sa;Password=sa;Initial Catalog=Morpheus\_BXF". To the right of this field is a "Test connection" button. Below this is a section for "Processing Agent Name" with a text field containing "BXFQuery". The next section is "BXF Schema Version for messages generated" with a dropdown menu set to "V1.0 2008" and a "Multi Thread" checkbox that is currently unchecked. The main area of the window is titled "Schedule/AsRun" and contains a tabbed interface. The tabs are "Translation", "Extras", "AsRun", and "Other translations". The "AsRun" tab is currently selected. Within this tab, there is a text field for "AsRun Connection String" and a dropdown menu for "AsRun Type" set to "BasicAsRun". At the bottom right of the window are "Cancel" and "Save" buttons.

Fig. 2-42: BXF Query Service Configuration Window - AsRun Tab

### - AsRun Connection String

Enter the connection string that will provide access to the AsRun database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxfl catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - AsRun Type

Default: BasicAsRun

For a received BXF Query AsRun message, from the drop-down menu select the level of detail to be included in the response from one of the following:

---

Note: The internal structures of the generated BXF messages differ greatly between the two options.

---

- **BasicAsRun**

The information included in the response message for each event equates to only that which conforms to the BXF standard:

- EventID
- Event status: 'Completed' or 'Failed'
- Actual start time
- Duration

- **CompleteAsRun**

The following information will be included in the response message for each event:

- All information included in the 'BasicAsRun'
- Any additional information that has been configured to be included in the 'PrivateInformation' element (refer to Schedule / AsRun: Extras Tab)

## Schedule / AsRun: Other Translations Tab

The BXF standard specifies three video / audio event transition rates: 'Fast', 'Medium', and 'Slow' - these are a representation of actual frame rates. Morpheus, however, processes actual frame rates only, and does not recognise the BXF names. In order for Morpheus to be compatible with the BXF standard, a translation mechanism is accessible from the 'Other Translations' configuration tab.

The screenshot shows the 'BKF Query Configuration' window with the 'Other translations' tab selected. The window contains the following fields and controls:

- BKF Repository Connection String:** A text field containing 'Data Source=172.31.230.102;User ID=sa;Password=sa;Initial Catalog=Morpheus\_BKF' and a 'Test connection' button.
- Processing Agent Name:** A text field containing 'BKFQuery'.
- BKF Schema Version for messages generated:** A dropdown menu set to 'V1.0 2008'.
- Multi Thread:** An unchecked checkbox.
- Translation / Extras / AsRun / Other translations:** A set of tabs with 'Other translations' selected.
- TransitionRate:** A section with a description: 'Transition rate is used the speed of a Video or Audio transition. In BKF exist 3 TransitionRate: Fast, Medium and Slow, Morpheus otherwise work with number of frames.' Below this are three time input fields:
  - Fast: 00:00:00:00
  - Medium: 00:00:00:00
  - Slow: 00:00:00:00
- Include events with Bxfiled zero (only for schedule queries):** A checked checkbox.
- Buttons:** 'Cancel' and 'Save' buttons at the bottom right.

Fig. 2-43: BKF Query Service Configuration Window - Other Translations Tab

### Transition Rate

#### **- Fast / Medium / Slow**

The TransitionRate fields allow specific Morpheus frame rates to be translated to BXF transition rates for entry into the response message to the external system. Enter an actual Morpheus frame rate for each of the BXF rates.

#### **- Include Events With Bxflid Zero (Only for Schedule Queries)**

Default: enabled (ticked).

Recommended setting: disabled.

When enabled, all events are itemised in a BXF response message, including those that do not have a unique Bxflid (i.e. those with a default Bxflid of all zeros).

In response to a query message with an identifier of 'Schedule' or 'AsRun', it is common practice to only itemise events with unique Bxflid's, typically the main events. Sub-events will normally have default Bxflid's of all zeros. With this feature enabled, all of these events will be included in the response message.

---

Note: If running BXF Synchronisation, this parameter must be disabled (un-ticked).

---



## Content (Item / Instance) Tab

The configurations on this tab are concerned with additional characteristics of the information that is to be transmitted in a BXF content query reply message by this service.

The screenshot shows the 'BXF Query Configuration' window with the 'Content (Item/Instance)' tab selected. The window has a blue title bar with the text 'HS BXF Query Configuration'. Inside, there are several sections: 'BXF Repository Connection String' with a text box containing 'Data Source=172.31.230.102;User ID=sa;Password=sa;Initial Catalog=Morpheus\_BXF' and a 'Test connection' button; 'Processing Agent Name' with a text box containing 'BXFQuery'; 'BXF Schema Version for messages generated' with a dropdown menu set to 'V1.0 2008' and a 'Multi Thread' checkbox; and a tabbed interface with 'Schedule/AsRun' and 'Content (Item/Instance)' tabs. The 'Content (Item/Instance)' tab is active, showing a checkbox for 'Generate message by Master Material ID', a 'Number of material to break the message' spinner set to 800, and a 'Device path' dropdown menu set to 'StorageLocation'. At the bottom right are 'Cancel' and 'Save' buttons.

HS BXF Query Configuration

BXF Repository Connection String Test connection

Data Source=172.31.230.102;User ID=sa;Password=sa;Initial Catalog=Morpheus\_BXF

Processing Agent Name

BXFQuery

BXF Schema Version for messages generated

V1.0 2008 Multi Thread

Schedule/AsRun **Content (Item/Instance)**

☐ Generate message by Master Material ID

Number of material to break the message

800

Device path

StorageLocation

Cancel Save

Fig. 2-44: BXF Query Service Configuration Window - Content (Item / Instance Tab)

### - Generate Message by Master Material ID

Default: disabled (un-ticked)

The MAPP database treats any Material ID as a single clip, therefore main material and soft segments are individually identified. This feature allows the service to generate a BXF message in one of the two following manners:

- Disabled: only the Material IDs of those clips that have changed will be entered into the message
- Enabled: the service scans the MAPP database for any clip that has the same Long\_File\_Id as the one that has changed, then generates a message containing the main Material ID and the IDs of any soft segments that relate to it

### - Number of Material to Break the Message

Default: 800 (recommended).

Set the maximum number of material items that can be included in a BXF Content Query Reply message before it is fragmented.

The BXF standard recommends that messages be limited to a maximum size of 2MBytes. An *All Items and Instances* 'Content' query response, by its nature, can be in excess of the imposed limit, in which case the BXF service must perform the following actions:

- Fragment the message
- Set the 'chunkingFlag' in the BXF message to 'True' in order to indicate that the message is being transmitted in multiple parts:
- `chunkingFlag = true`
- Indicate the number of each message relative to the others (e.g. 2 of 10):  
`itemNumber = 'positive integer' (the number of the message)`  
`totalItems = 'positive integer' (the total number of messages)`

This BXF recommendation has only been applied to the BXF Content Query Reply message of this service.

All Host Shell BXF services will accept messages greater than 2MBytes in size, and it is reasonable to expect that external BXF services will do the same.

### - Device Path

From the drop-down menu, select the format of the path that will be used in a generated message in order to define the device on which the changed content is stored. The format indicates the MAPP database 'Server Type' field from which the values will be extracted in order to define the path. Select from the following options:

- **StorageLocation:** concatenate the value in the DEVICE\_NO field with that of the STORAGE\_LOCATION field in order to create the following path format:  
`\\DEVICE_NO/STORAGE_LOCATION`
- **StorageLocationUNC:** use the value in the STORAGE\_LOCATION\_UNC field only, in order to create the following path format:  
`\\STORAGE_LOCATION_UNC`

## BXF Schedule Change Notifier Service

Upon receipt of schedule change notifications directly from the EventStore, this service will generate a BXF message for transmission to preconfigured destinations. The service receives notifications from the EventStore for all channels.

The BXF message generated is of type 'Data', and with the identifier 'Schedule'.

When any BXF Synchronisation mode is configured, if this service is acting as the Source, it will receive a message from the Source BXF Schedule Service requesting that it be added / removed as a destination.

BXF Synchronisation is documented in the BXF Implementation Configuration document.

### BXF Schedule Change Notifier Service: Principle of Operation

The functionality of the BXF Schedule Change Notifier service is shown in Figure 2-45 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

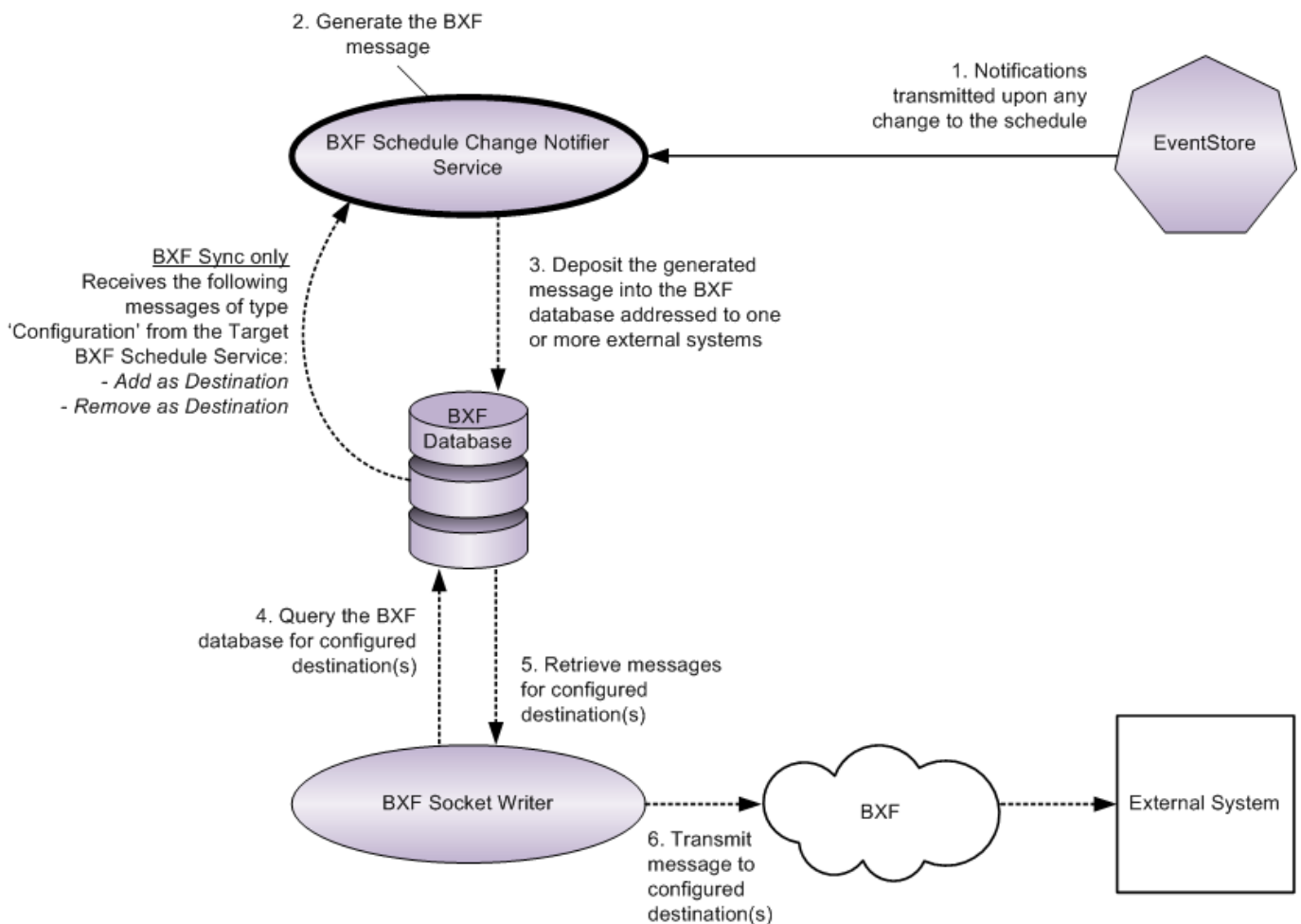
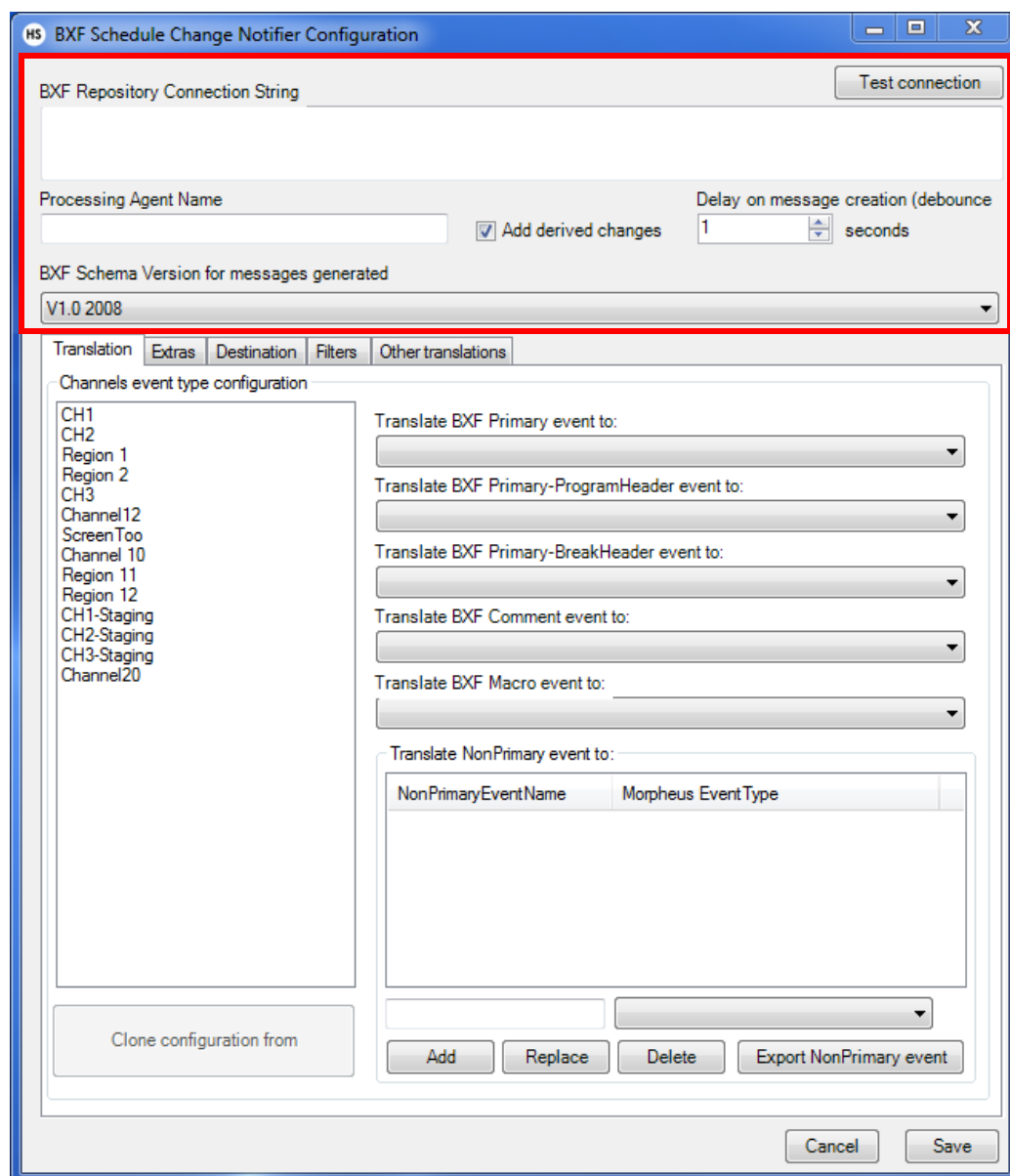


Fig. 2-45: BXF Schedule Change Notifier Service Logical Topology

## Common Parameters

The following parameters are common to all tabs in the BXF Schedule Change Notifier Service Configuration window (Figure 2-46).



The image shows the 'BXF Schedule Change Notifier Configuration' window with the 'Translation' tab selected. A red rectangle highlights the top section containing common parameters. Below this, the 'Channels event type configuration' section is visible, showing a list of event types on the left and translation options on the right.

**Common Parameters (highlighted in red):**

- BXF Repository Connection String: [Text Field] [Test connection]
- Processing Agent Name: [Text Field]
- Delay on message creation (debounce): [1] seconds
- ☒ Add derived changes
- BXF Schema Version for messages generated: V1.0 2008

**Translation Tab:**

- Translation | Extras | Destination | Filters | Other translations
- Channels event type configuration
- CH1, CH2, Region 1, Region 2, CH3, Channel12, ScreenToo, Channel 10, Region 11, Region 12, CH1-Staging, CH2-Staging, CH3-Staging, Channel20
- Translate BXF Primary event to: [Dropdown]
- Translate BXF Primary-ProgramHeader event to: [Dropdown]
- Translate BXF Primary-BreakHeader event to: [Dropdown]
- Translate BXF Comment event to: [Dropdown]
- Translate BXF Macro event to: [Dropdown]
- Translate NonPrimary event to: [Table]

NonPrimaryEventName	Morpheus EventType

[Clone configuration from] [Add] [Replace] [Delete] [Export NonPrimary event] [Cancel] [Save]

Fig. 2-46: BXF Schedule Change Notifier Service Configuration Window - Translation Tab

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxfl catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalog=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Processing Agent Name

Enter a name that will identify the BXF Schedule Change Notifier Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Processing Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message to the external system, and as the Destination System ('destination' field entry of the Messages table in the BXF database) or receiving messages from the external system.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### - Add Derived Changes

Default: enabled.

Recommended setting: disabled.

When enabled, indirect changes to an event are reported in a schedule change message from this service. The event to which a direct change is made will always be reported in the message, regardless of the setting of this feature.

If this function is disabled, then only direct changes are reported in a schedule change message from this service.

### Example

If the duration of an event is changed directly, the start time of follow-on events will change indirectly - the change to the follow-on events (the 'derived' change) will only be reported in the message if this feature is enabled.

---

Note: This is a service-wide function that will apply to all channels in the system, regardless of whether or not they are enabled on the 'Filters' tab (refer to Filters Tab).

---

---

Note: If enabled, the events reported in a schedule change message will be refined in accordance with the configurations on the 'Filters' tab (refer to Filters Tab).

---

---

Note: This function must be enabled in order for the following feature on the 'Filters' tab to operate (Filters Tab):

- 'Ignore Derived Changes Caused by Channel in Rippling Hold'
- 

---

Note: This feature must be disabled if this instance of the service is configured for a Multi-Site Sync environment.

---

### - Delay On Message Creation (Debounce)

The default is 1 second.

Recommended setting: 5 seconds.

The interval in seconds between the creation of schedule change messages by this service. The configured delay starts at the point at which the previous message creation process had finished.

Regardless of the value attributed to this function, the time required to transmit a message from source to destination is approximately 5 seconds.

---

Note: If configuring an instance of this service in a Multi-Site Sync environment, this feature should be set to 1 second (the default).

---

## Translation Tab

Channel specific parameters.

The BXF standard defines a number of default event types to which specific Morpheus event types must be matched in order for the service to build the BXF Schedule Change message with the requested information. The matching process is known as a 'Translation'.

Note: Configurations are applied on a per channel basis.

**BXF Schedule Change Notifier Configuration**

BXF Repository Connection String Test connection

Processing Agent Name ☒ Add derived changes Delay on message creation (debounce) 1 seconds

BXF Schema Version for messages generated V1.0 2008

**Translation** Extras Destination Filters Other translations

Channels event type configuration

- CH1
- CH2
- Region 1
- Region 2
- CH3
- Channel12
- ScreenToo
- Channel 10
- Region 11
- Region 12
- CH1-Staging
- CH2-Staging
- CH3-Staging
- Channel20

Clone configuration from

Translate BXF Primary event to:

Translate BXF Primary-ProgramHeader event to: Not supported - refer to parameter description

Translate BXF Primary-BreakHeader event to:

Translate BXF Comment event to: Not supported - refer to parameter description

Translate BXF Macro event to: Not supported - refer to parameter description

Translate NonPrimary event to:

NonPrimaryEventName	Morpheus EventType

A B

Add Replace Delete Export NonPrimary event

Cancel Save

Fig. 2-47: BXF Schedule Change Notifier Service Configuration Window - Translation Tab

### - Channels Event Type Configuration List

The list of Morpheus channels to which translations can be applied. This menu is populated automatically from the EventStore.

### - Clone Configuration From

Apply the set of existing primary and non-primary event translations on one channel to another:

- a) Select the channel on which to apply the set of translation.
- b) Click on the **Clone Configuration From** Button.
- c) The 'Select Source Channel' window is displayed (Figure 2-48).

From the drop-down menu, select the channel from which the translations will be cloned, then click on **Ok**.

The translation configuration is applied to the selected channel.

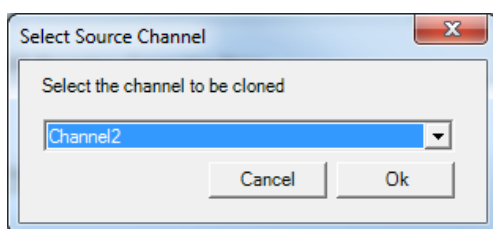


Fig. 2-48: Select Source Channel

---

Note: The receiving channel must support the event type related to the translation that is being cloned.

---

---

Note: This feature identifies whether or not the source event translations include the source channel name, and if so, that portion is changed to the name of the target channel during the cloning process.

---

### - Translate BXF Primary Event To

No default.

From the drop-down menu select the Morpheus event type that will translate to the BXF Primary Event type.

### - Translate BXF Primary-ProgramHeader Event To

This BXF event type is not currently supported by the BXF Schedule Change Notifier Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---



**- Translate BXF Primary-BreakHeader Event To**

No default.

From the drop-down menu select the Morpheus event type that will translate to the BXF Primary-BreakHeader Event.

**- Translate BXF Comment Event To**

This BXF event type is not currently supported by the BXF Schedule Change Notifier Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

**- Translate BXF Macro Event To**

This BXF event type is not currently supported by the BXF Schedule Change Notifier Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

### - Translate NonPrimary Event To

No defaults.

Non-primary event names (e.g. sub-events for GPIs and subtitles) are not specified in the BXF standard, and as such they are user definable.

---

Note: Any NonPrimaryEventName configured for this feature must match with that configured on the system from which messages will be received and responses transmitted.

---

- **Add**

- a) Click on a channel in order to select it.
- b) In the text box 'A' (Figure 2-47), enter a name for the non-primary event that will be sent in a BXF message.

---

Note: Names are case sensitive.

---

- c) From the drop-down menu 'B' (Figure 2-47), select the Morpheus event type that will translate to the non-primary event selected in box 'A'.
- d) Click on the **Add** button - the new translation entry is added to the table.

- **Replace**

To modify an existing translation entry, click on any part of it and the text box and drop-down menus will be populated with its current properties. Enter alternative properties then click on the **Replace** button.

- **Delete**

To remove a translation entry from the table, click on any part of it and click on the **Delete** button.

### - Export NonPrimary Event

Save to a file the non-primary event translations configured for a selected channel. This function is only provided for administrative records - there is no import capability to allow sharing of the configured translations.

1. From the channel list, select the one for which to export the translations.
2. Click on the **Export NonPrimary Event** button.
3. Windows Explorer is opened: enter a filename and destination folder, then click on the **Save** button.

---

Note: Only the non-primary events for the currently selected channel are exported.

---

## Extras Tab

If additional event parameter fields and their values, that are not standard BXF elements, are to be defined for inclusion in the 'PrivateInformation' element of the BXF message, then these should be defined on the BXF Schedule Change Notifier Service **Extras** Tab, as shown in Figure 2-49.

Note: The 'PrivateInformation' element of the BXF message is supported in all BXF schemas.

The screenshot shows the 'BKF Schedule Change Notifier Configuration' window with the 'Extras' tab selected. The window contains the following elements:

- BKF Repository Connection String:** A text input field with a 'Test connection' button.
- Processing Agent Name:** A text input field.
- Delay on message creation (debounce):** A numeric input field set to '1' with a 'seconds' label.
- BKF Schema Version for messages generated:** A dropdown menu showing 'V1.0 2008'.
- Tabs:** 'Translation', 'Extras' (selected), 'Destination', 'Filters', and 'Other translations'.
- Hierarchy level of the event status to be exported:** A dropdown menu showing 'None'.
- Define which fields from Event.Fields.Parameter must go to BKF private section:**
  - Parameters to be added:** A list box with a header 'Parameter name'.
  - Enter event parameter field here...:** A text input field highlighted with a red border.
  - Buttons:** 'Add', 'Replace', and 'Delete' buttons.
- Bottom Buttons:** 'Cancel' and 'Save' buttons.

Fig. 2-49: BKF Schedule Change Notifier Service Configuration Window - Extras Tab

### **- Hierarchy Level of the Event Status to be Exported**

The severity level of event status log entries that are to be included in the BXF messages. Only those entries that have been generated at the severity level selected will be transmitted in a BXF message. Entries will be included in the 'PrivateInformation' element of the BXF message.

From the drop down menu select the required message severity level. The selected hierarchy level will apply to event status entries for all channels that are selected to use this service.

---

Note: The Hierarchy Level of the Event Status to be Exported parameter is of no relevance to the configuration of the BXF Synchronisation process.

---

Select from the following:

- None
- Information
- Warning
- Error
- Critical

**- Parameters to be Added**

- **Add**

- a) Enter the event parameter field to include in the BXF message, as defined in the event's **All Parameter Definitions** tab in the Configurator.
- b) Click on **Add**.

---

Note: Parameters are case sensitive.

---

- **Replace**

Highlight the event parameter field name entry to be changed - the text entry box will be populated with the entry. Enter the new event parameter field name and click on **Replace**.

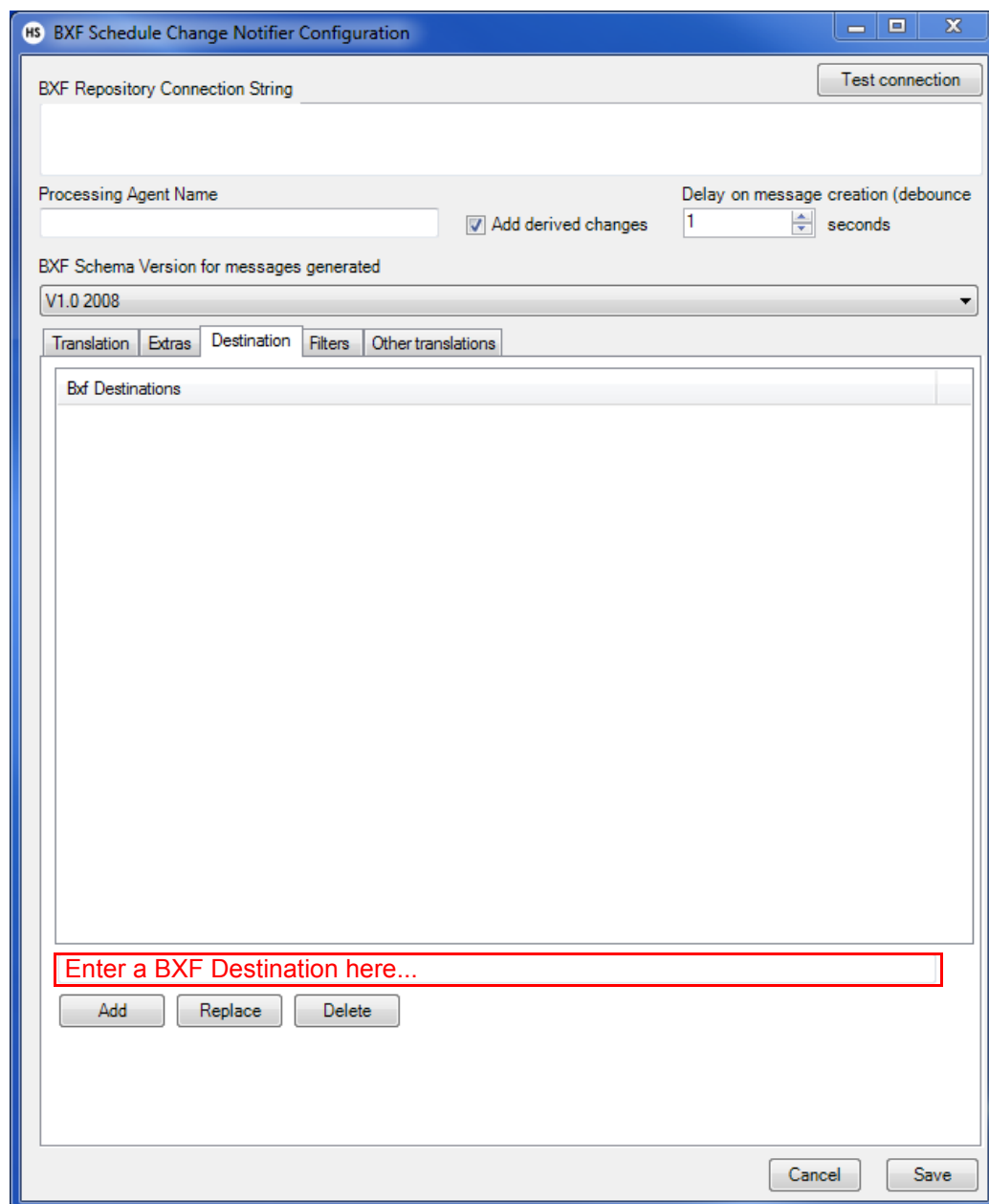
- **Delete**

Highlight the event parameter field name to delete then click on **Delete**.

## Destination Tab

The names of those services on the Target system(s) that are to receive BXF messages from this service must be configured on the 'Destination' tab of the BXF Schedule Change Notifier service. The name of the destination service is supplied to the BXF Socket / File Writer for forwarding.

Note: When this service is configured in a 'Multi-Site Sync' environment, it transmits messages to the 'BXF Schedule Service' on the opposite site.



The image shows a screenshot of the 'BXF Schedule Change Notifier Configuration' window, specifically the 'Destination' tab. The window has a title bar with 'HS BXF Schedule Change Notifier Configuration' and standard window controls. Inside, there's a 'BXF Repository Connection String' field with a 'Test connection' button. Below that is a 'Processing Agent Name' field, a checked 'Add derived changes' checkbox, and a 'Delay on message creation (debounce)' spinner set to '1' seconds. A 'BXF Schema Version for messages generated' dropdown is set to 'V1.0 2008'. The 'Destination' tab is selected among 'Translation', 'Extras', 'Destination', 'Filters', and 'Other translations'. The main area is titled 'Bxf Destinations' and is currently empty. At the bottom of this area is a red-bordered text box with the placeholder 'Enter a BXF Destination here...'. Below the text box are 'Add', 'Replace', and 'Delete' buttons. At the very bottom of the window are 'Cancel' and 'Save' buttons.

Fig. 2-50: BXF Schedule Change Notifier Service Configuration Window - Destination Tab

### - BXF Destinations

- **Add**

- a) In the text box, enter the name of the destination service to which messages are to be sent.

---

Note: Parameters are case sensitive.

---

- b) Click on **Add** - the destination service is added to the 'BXF Destinations' pane.

- **Replace**

Highlight the destination entry to be changed - the text entry box will be populated with the entry. Enter the new destination and click on **Replace**.

- **Delete**

Highlight the destination entry to delete then click on **Delete**.

## Filters Tab

Channel specific filters that can be applied independently in order to refine the conditions under which an event is reported in a schedule change message.

All filters are applied collectively.

The service interrogates the EventStore in order to identify changes to events.

Note: Some filters are service-wide and are indicated as such in Figure 2-51.

The screenshot shows the 'BXF Schedule Change Notifier Configuration' window with the 'Filters' tab selected. The 'Service-wide filters' section is highlighted with a red box, containing two checkboxes: 'Ignore derived changes caused by Channel in rippling hold' and 'Ignore duration changes on Program Event and children for Channel in rippling hold'. Below this, the 'Per-channel filters' section lists 'Enabled channels' including CH1, CH2, Region 1, Region 2, CH3, Channel12, ScreenToo, Channel 10, Region 11, Region 12, CH1-Staging, CH2-Staging, CH3-Staging, Channel14, and Channel20. The 'Actions' section includes checkboxes for 'Events added', 'Events updated', and 'Events removed'. The 'Top level event types' section has tabs for 'Main event types', 'Sub-event types', and 'Fields', with 'Main event types' selected. It includes a checkbox for 'All top level event types' and a list of event types with 'Add', 'Replace', and 'Delete' buttons.

Fig. 2-51: BXF Schedule Change Notifier Service - Filters Tab



### **- Ignore Derived Changes Caused by Channel in Rippling Hold**

---

Note: A service-wide filter that is applied to all channels (the 'Enabled Channels' list in the 'Per Channel Filters' pane is unrelated to this function).

---

Default: disabled.

When enabled, this service will determine if changes to the start time of events are due to the channel being in 'Rippling Hold', and if so will not report those events in a schedule change message. The program event will be reported on unless the 'Ignore Duration Changes on Program Event and Children for Channel in Rippling Hold' is enabled (below).

---

Note: By default, the service-wide function 'Add Derived Changes' is enabled and therefore any change to the start time of an event will be reported in a message, whether a direct change (e.g. by an operator), or a derived change.

The service-wide function 'Add Derived Changes' must be enabled in order for this filter to be effective (refer to Common Parameters).

---

### **- Ignore Duration Changes on Program Event and Children for Channel in Rippling Hold**

---

Note: A service-wide filter that is applied to all channels (the 'Enabled Channels' list in the 'Per Channel Filters' pane is unrelated to this function).

---

Default: disabled.

When enabled, this service will determine if changes to the duration of the program event is due to it being held, and if so will not report that event, or its children, in a schedule change message.

---

Note: This function was designed to be supplement 'Ignore Derived Changes Caused by Channel in Rippling Hold', although it is effective regardless of whether or not 'Add Derived Changes' is enabled.

---

#### Per Channel Filters Pane

##### **- Enabled Channels**

Default: all channels deselected.

Tick the checkbox in order to activate filtering on a per channel basis.

---

Note: Filter configuration persists if the channel on which they are configured is subsequently disabled on the 'Filters' tab.

---

---

Note: If enabled, the following service-wide filters will apply to all channels, regardless of whether or not they are enabled in this list:

- 'Ignore Derived Changes Caused by Channel in Rippling Hold'
  - 'Ignore Duration Changes on Program Event and Children for Channel in Rippling Hold'
- 

##### **- Actions: Events Added**

Default: enabled.

Only when this function is enabled will an event that has been added to the schedule be reported in a schedule change message.

##### **- Actions: Events Updated**

Default: enabled.

Only when this function is enabled will an event that has been updated in the schedule be reported in a schedule change message.

##### **- Actions: Events Removed**

Default: enabled.

Only when this function is enabled will an event that has been removed from the schedule be reported in a schedule change message.

#### Per Channel Filters Pane

##### **- Ignore Events Removed After Been Played**

Default: disabled.

The standard behaviour of the service is for a new message to be generated upon any change to the schedule, including when an event has been AsRun collected, having already played out.

Tick the checkbox in order to stop this service from reporting on events when they are removed from the schedule, having already played out.

Applied on a per-channel basis.

##### **- Update Messages Only Uses Fields Filter**

Default: disabled.

When enabled, this function deactivates all filters apart from those on the 'Fields' tab (detailed below).

Applied on a per-channel basis, and only for 'update' messages.

##### **- Within <n> TOP LEVEL Events**

Default: 0

Limit the service to reporting on the top <n> number of events in the schedule.

When the value is 0, all events in the schedule are reported on.

Applied on a per-channel basis.

### Per Channel Filters Pane: Top Level Event Types Tab

A filter to limit the service to report only on a specific selection of top level event types.  
Applied on a per-channel basis.

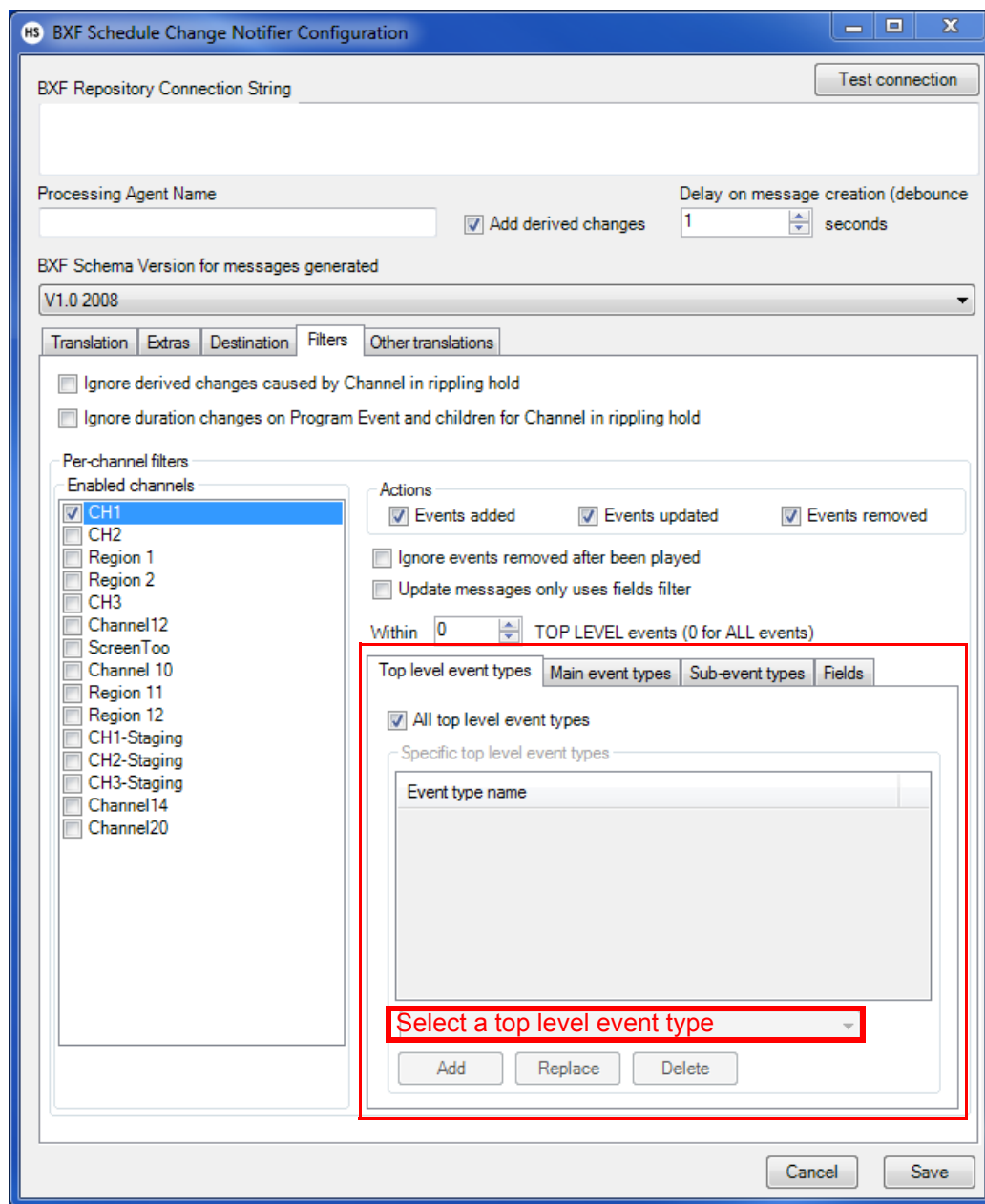


Fig. 2-52: BXF Schedule Change Notifier Service - Filters Tab - Top Level Event Types

### **- All Top Level Event Types**

Default: enabled.

When enabled, the service reports on all top level event types in the schedule, in which case the 'Specific Top Level Event Types' configurations are deactivated and the pane is greyed out.

### - Specific Top Level Event Types

Configure the service to report changes only for those top level event types that have been specified.

---

Note: 'All Top Level Event Types' must be disabled.

---

- **Add**

- a) From the drop-down menu, select a top level event type.

---

Note: The service interrogates the EventStore in order to populate the drop-down menu with the top level event types configured on the channel.

---

- b) Click on **Add**. The top level event type is added to the 'Specific Top Level Event Types' list.

- **Replace**

To modify an existing top level event type, click to select it and it will appear in the text box. Select another top level event type from the drop down menu then click on the **Replace** button.

- **Delete**

To remove a top level event type, click to select it then click on the **Delete** button.

### Per Channel Filters Pane: Main Event Types Tab

A filter to limit the service to report only on a specific selection of main event types.  
Applied on a per channel basis.

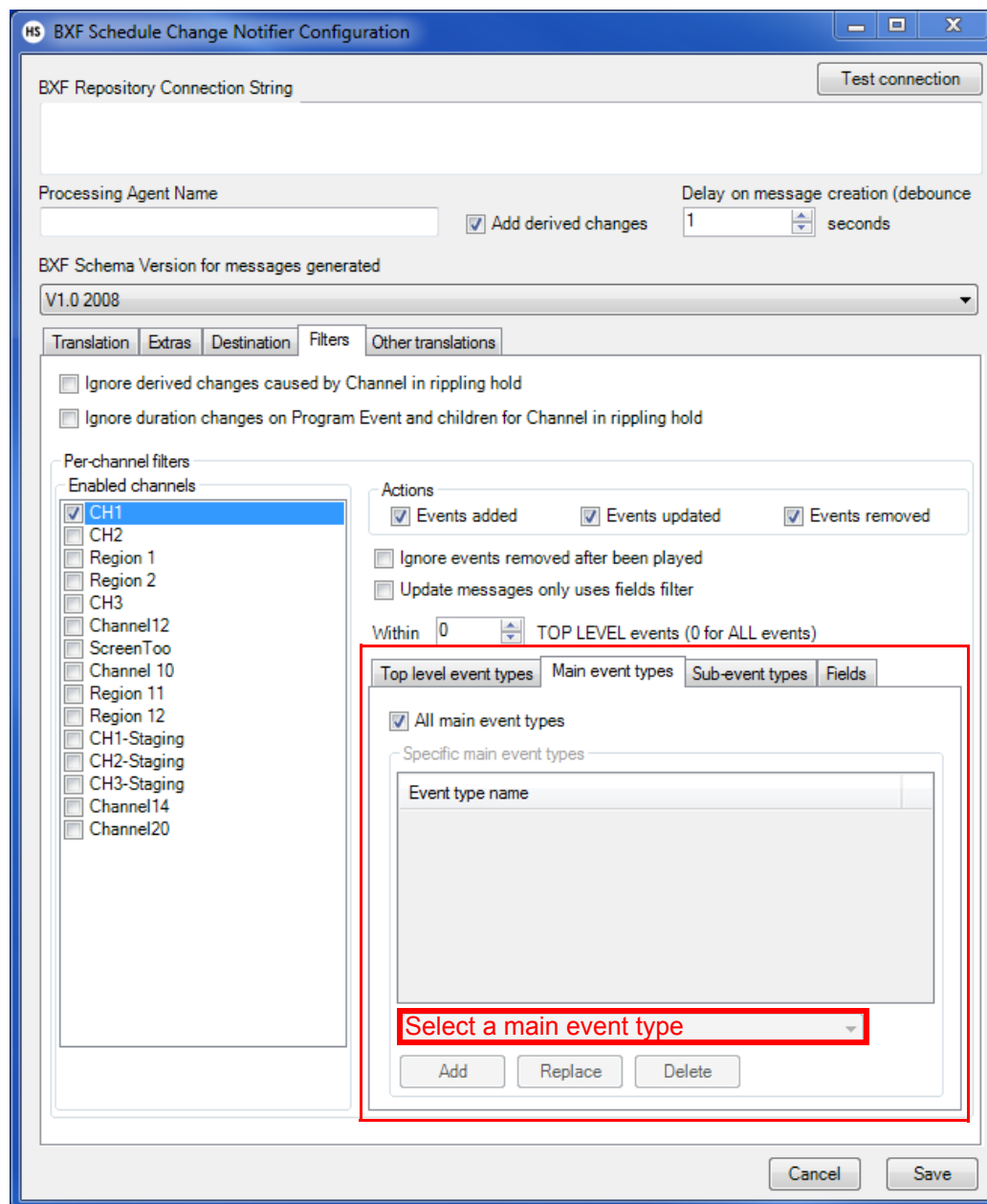


Fig. 2-53: BXF Schedule Change Notifier Service - Filters Tab - Main Event Types

### **- All Main Event Types**

Default: enabled.

When enabled, the service reports on all main event types in the schedule - in which case the 'Specific Main Event Types' configurations are deactivated and the window is greyed out.

### - Specific Main Event Types

Configure the service to report changes only for those main event types that have been specified.

---

Note: 'All Main Event Types' must be disabled.

---

- **Add**

- a) From the drop-down menu, select a main event type.

---

Note: The service interrogates the EventStore in order to populate the drop-down menu with the main event types configured on the channel.

---

- b) Click on **Add**. The selected main event type is added to the 'Specific Main Event Types' list.

- **Replace**

To modify an existing main event type, click to select it and it will appear in the text box. Select another main event type from the drop down menu then click on the **Replace** button.

- **Delete**

To remove a main event type, click to select it then click on the **Delete** button.

### Per Channel Filters Pane: Sub-Event Types Tab

A filter to limit the service to report only on a specific selection of sub-event types.  
Applied on a per channel basis.

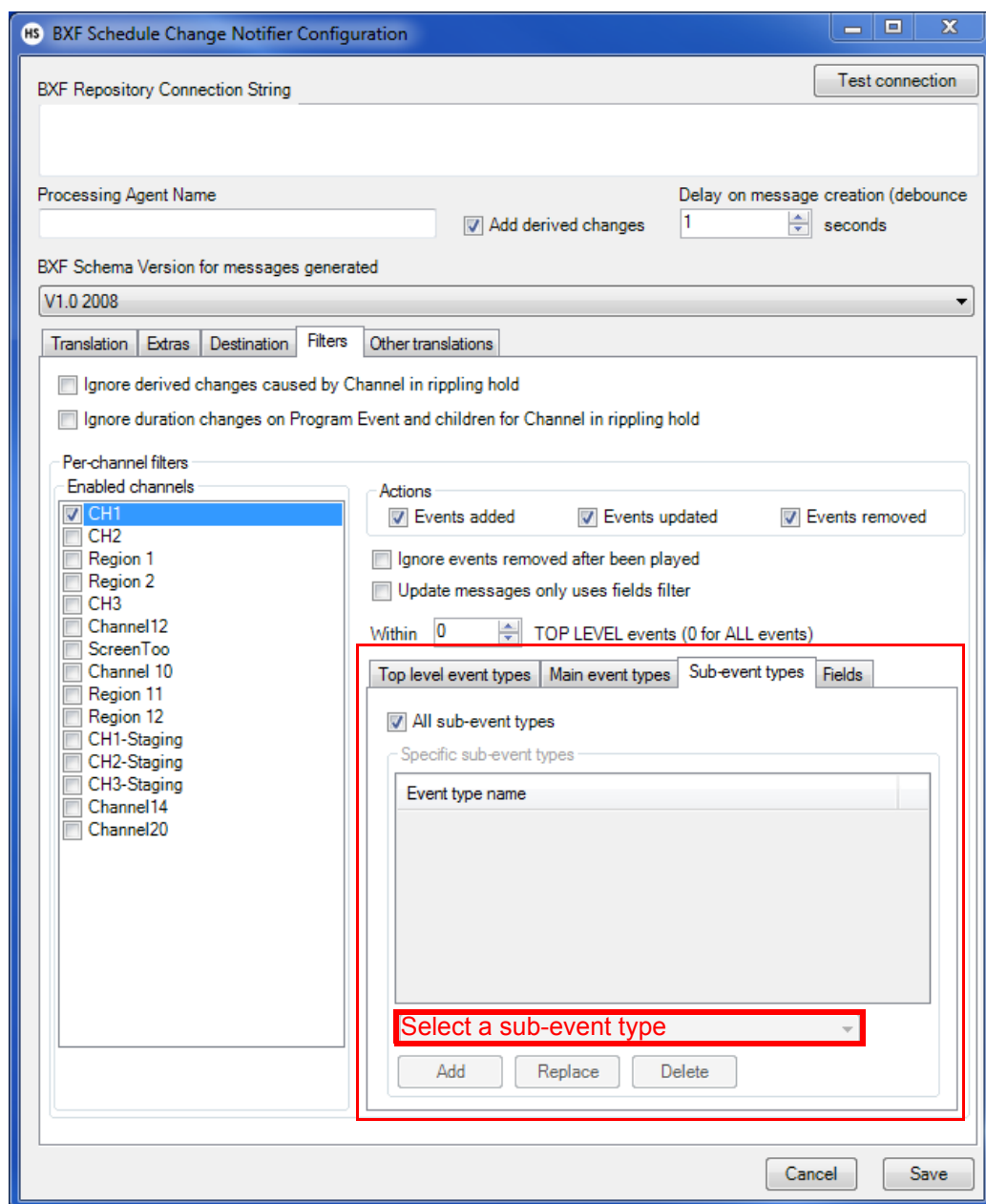


Fig. 2-54: BXF Schedule Change Notifier Service - Filters Tab - Sub-Event Types

### **- All Sub-Event Types**

Default: enabled.

When enabled, the service reports on all sub-event types in the schedule - in which case the 'Specific Sub-Event Types' configurations are deactivated and the window is greyed out.



### - Specific Sub-Event Types

Configure the service to report changes only for those sub-event types that have been specified.

---

Note: 'All Main Event Types' must be disabled.

---

- **Add**

- a) From the drop-down menu, select a sub-event type.

---

Note: The service interrogates the EventStore in order to populate the drop-down menu with the sub-event types configured on the channel.

---

- b) Click on **Add**. The selected sub-event type is added to the 'Specific Sub-Event Types' list.

- **Replace**

To modify an existing sub-event type, click to select it and it will appear in the text box. Select another sub-event type from the drop down menu then click on the **Replace** button.

- **Delete**

To remove a sub-event type, click to select it then click on the **Delete** button.

### Per Channel Filters Pane: Fields Tab

A filter to limit the service to report only on a specific selection of event parameter fields if they are updated in the EventStore.

Applied on a per channel basis.

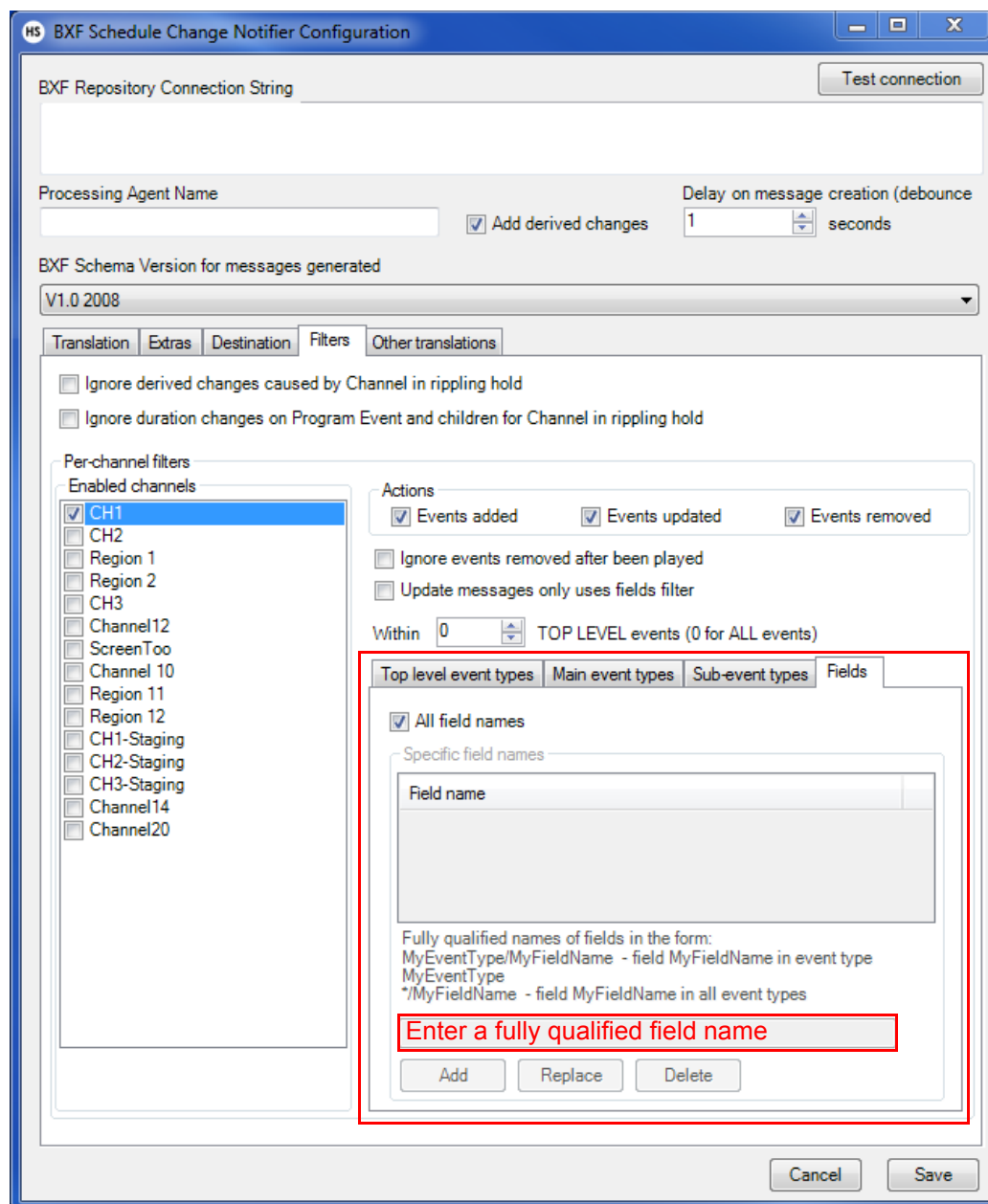


Fig. 2-55: BXF Schedule Change Notifier Service - Filters Tab - Fields

### - All Field Names

Default: enabled.

When enabled, the service reports on all event fields in the schedule - in which case any 'Specific Field Names' configurations are deactivated and the window is greyed out.

### - Specific Field Names

Configure the service to only report updates for the event fields that have been specified.

---

Note: 'All Field Names' must be disabled.

---

Specify fully qualified field names as shown in the following examples:

*EventTypeX/FieldNameY*

The service reports only upon an update to *FieldNameY* of *EventTypeX*.

*\*/FieldNameY*

The service reports upon an update to *FieldNameY* for any event type.

- **Add**

- a) Enter the fully qualified field name into the text box.

---

Note: Field names are case sensitive.

---

- b) Click on **Add** in order to save the fully qualified field name to the list of 'Specific Field Names'.

- **Replace**

To modify an existing fully qualified field name, click to select it and it will appear in the text box. Modify the fully qualified field name then click on the **Replace** button.

- **Delete**

To remove a fully qualified field name, click to select it and click on the **Delete** button.

## Other Translations Tab

The BXF standard specifies three video / audio event transition rates: 'Fast', 'Medium', and 'Slow' - these are a representation of actual frame rates. Morpheus, however, processes actual frame rates only, and does not recognise the BXF names. In order for Morpheus to be compatible with the BXF standard, a translation mechanism is accessible from the 'Other Translations' configuration tab.

The screenshot shows the 'BXF Schedule Change Notifier Configuration' window with the 'Other translations' tab selected. The window contains the following fields and controls:

- BXF Repository Connection String:** A text input field with a 'Test connection' button to its right.
- Processing Agent Name:** A text input field.
- Add derived changes:** A checked checkbox.
- Delay on message creation (debounce):** A numeric input field set to '1' with 'seconds' as the unit.
- BXF Schema Version for messages generated:** A dropdown menu showing 'V1.0 2008'.
- Tabbed Interface:** Tabs for 'Translation', 'Extras', 'Destination', 'Filters', and 'Other translations' (which is active).
- TransitionRate Section:**
  - A descriptive text: 'Transition rate is used the speed of a Video or Audio transition. In BXF exist 3 TransitionRate: Fast, Medium and Slow, Morpheus otherwise work with number of frames.'
  - Three time input fields for 'Fast', 'Medium', and 'Slow', each currently set to '00:00:00:00'.
- Buttons:** 'Cancel' and 'Save' buttons at the bottom right.

Fig. 2-56: BXF Schedule Change Notifier Service - Other Translations Tab

### Transition Rate

#### **- Fast / Medium / Slow**

The TransitionRate fields allow specific Morpheus frame rates to be translated to BXF transition rates for entry into a message for the external system. Enter an actual Morpheus frame rate for each of the BXF rates.

## BXF Schedule Service

A service that receives schedules in a BXF format from which it will update the EventStore. It processes messages for the following:

- Adding a new schedule
- Deleting an existing schedule
- Adding events to an existing schedule
- Deleting events from an existing schedule
- Updating events in an existing schedule

Any schedule updates from the BXF Schedule Change Notifier Service are ignored until the full schedule has been received from the BXF Query Service.

- When any BXF Synchronisation mode is configured, if this service is acting as the Target, it will send a message to the Source BXF Schedule Change Notifier Service requesting that it be added as a destination, and a message to the Source BXF Query Service requesting a complete schedule.  
BXF Synchronisation is documented in the BXF Implementation Configuration document.

It is also the responsibility of the 'BXF Schedule Service' to evaluate the presence of BXF locks on main events and determine whether or not changes should be applied from a received BXF message. If the 'BxfLocked' field is set to TRUE on a main event then it will not be updated - if the 'BxfLocked' field is set to FALSE, then the main event will be changed accordingly.

In this respect:

- The response to a received message containing only changes to locked events will be a NAK, the schedule will not be modified, and a warning is posted in the validation column of the editor against the locked events in order to inform the operator that the received updates were not applied.
- The response to a message containing both changes to locked and to non-locked events will be an ACK, but changes will only be applied to non-locked events. A warning is posted in the validation column of the editor against the locked events in order to inform the operator that the received updates were not applied.

## BXF Schedule Service: Principle of Operation

The functionality of the BXF Schedule service is shown in Figure 2-57 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

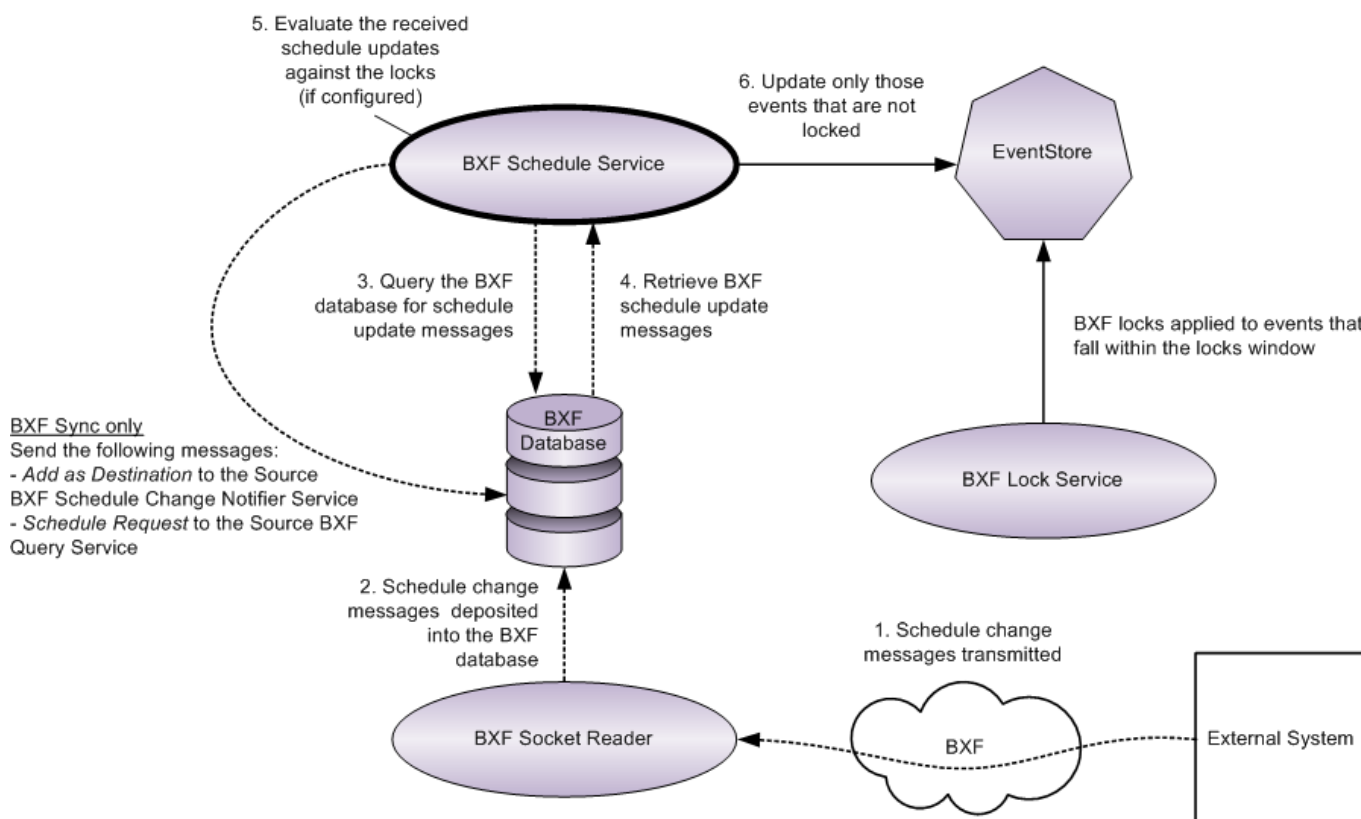


Fig. 2-57: BXF Schedule Service Logical Topology

## BXF Schedule Configuration - Common Parameters

Service-wide parameters for an instance of the service.

HS BXF Schedule Configuration

BXF Repository Connection String Test connection  
Data Source=172.31.230.102;User ID=sa;Password=sa;Initial Catalog=Morpheus\_BXF

Processing Agent Name: BXF Schedule Loader  
Close to Air Safety Limit: 1 minutes  
Margin for the on air event: 10 seconds  
Maximum number of subevents: 60  
BXF Schema Version for messages generated: V1.0 2008  
☐ Do not load schedules containing Primary events with zero durations  
Order to process the messages: GeneratedDateTime  
Language for attributes with language option: eng  
☒ Store processed message and SCH schedule when possible  
Path to store the processed message: C:\BXFScheduleMessagesLog  
Path to store the processed message with errors: C:\BXFScheduleMessagesLog

Channels event type configuration | Messages special actions | Offline editor mode & Export .sch | Other translation configuration | Warning report c

CH1  
CH2  
Region 1  
Region 2  
CH3  
Channel12  
ScreenToo  
Channel 10  
Region 11  
Region 12  
CH1-Staging  
CH2-Staging  
CH3-Staging  
Channel14  
Channel20

Translate BXF Primary event to:  
Translate BXF Primary-ProgramHeader event to:  
Translate BXF Primary-BreakHeader event to:  
Translate BXF Comment event to:  
Translate BXF Macro event to:  
☐ Accept events in the past

Translate NonPrimary event to:  
☐ Ignore NonPrimary events that are not mapped

NonPrimaryEventName	Morpheus EventType	Special Action

Clone configuration from: Add Replace Delete Export NonPrimary event

Cancel Save

Fig. 2-58: BXF Schedule Configuration

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Enter the connection string that will provide access to the AsRun database that this service will query for the existence of new records. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxfl catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Processing Agent Name

Enter a name that will identify the BXF Schedule Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Processing Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message to the external system, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages from the external system.

---

Note: It is recommended that the agent name is different to the instance name.

---



### - Close to Air Safety Limit

When set, a whole schedule will be rejected if it contains an event that is due to start within the configured threshold.

Enter a value in minutes:

- Default: 0
- Recommended: 5

By default, the BXF Schedule Service validates all messages before applying them to the EventStore, and if one event is in the past then the whole schedule is rejected - this feature is as an additional validation qualifier.

---

WARNING: This parameter is overridden by the enabling of **Ignore Close to air Safety Limit but do not Load Events on the Past** (refer to - Ignore Close to Air Safety Limit but do not Load Events on the Past).

---

### - Margin for the On Air Event

This parameter is synonymous with a preroll.

If a BXF message is received that contains an amendment to the on-air schedule, any events that are wholly in the past are rejected, and the replacement to the on-air event will only play out once the value assigned to this parameter has expired.

- Default: 10 seconds (this is also the minimum value).

---

Note: The parameter **Accept Events in the Past** must be ticked in order for this feature to be enabled - refer to Channels Event Type Configuration Tab.

---

### - Maximum Number of Subevents

If a main event, that is included in a received schedule message, contains more subevents than is assigned as the value of this parameter, then the message is rejected. Commercials are also considered to be subevents in this regard.

Enter a value representing the maximum number of permitted subevents:

- Default: 60
- Minimum: 10
- Maximum 80

This feature is for EventStore performance tuning.

#### **- BXF Schema Version for Messages Generated**

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

#### **- Do Not Load Schedules Containing Primary Events With Zero Duration**

Reject any received message that contains a schedule within which are zero duration primary events.

The default is disabled (un-ticked).

#### **- Language for Attributes with Language Option**

A BXF message may contain options in order for the 'Title' field of an event to be displayed in the Editor according to the local national language.

Enter the language code required as defined in the ISO 639-2 standard (Codes for the Representation of Names of Languages).

The default is 'eng' (English).

---

Note: The language field is required as of version 2 of the BXF standard.

---

#### **- Order to Process the Messages**

Select one of the following options:

- GeneratedDateTime

Messages are processed in the order in which they were created (the default).

- ScheduleStartDateTime

Messages are processed according to the start time of the schedule contained therein. This option allows scheduled events, that are close to being on-air, to be prioritised for processing.

### **- Store Processed Message and SCH Schedule When Possible**


Checkbox. The default is enabled (ticked).

If enabled, the procedure when receiving a message containing a schedule is to attempt to apply the schedule directly to the EventStore.

- If successful (i.e. '...When Possible' from the feature name), the original BXF message is then converted to a .sch file, and both the original message and the .sch file are stored in a zip file in the folder selected for 'Path to Store the Processed Message' (below).
- If the EventStore rejects the schedule due to errors, then no .sch file is created, and only the original BXF message is stored in the folder selected for 'Path to Store Processed Message with Errors' (below).


### **- Path to Store the Processed Message**

The folder in which to store the received message if processed successfully. The original message and the .sch are combined into a zip file.

The default folder is `C:\BXFScheduleMessagesLog`. Use the ellipsis button  to select an alternative folder.

### **- Path to Store the Processed Message with Errors**

The folder in which to store the received message if processing was unsuccessful due to erroneous content. Only the original BXF message is stored.

The default folder is `C:\BXFScheduleMessagesLog`. Use the ellipsis button  to select an alternative folder.

## Channels Event Type Configuration Tab

Channel specific parameters.

The BXF standard defines a number of default event types that must be matched to specific Morpheus event types prior to being loaded into the EventStore. The matching process is known as a 'Translation'.

Note: Configurations are applied on a per channel basis.

The screenshot shows the 'Channels event type configuration' tab. On the left is a list of channels: CH1, CH2, Region 1, Region 2, CH3, Channel12, ScreenToo, Channel 10, Region 11, Region 12, CH1-Staging, CH2-Staging, CH3-Staging, and Channel20. Below this list is a 'Clone configuration from' button. The main area contains several translation configuration sections:

- Translate BXF Primary event to:** A dropdown menu.
- Translate BXF Primary-ProgramHeader event to:** A dropdown menu showing 'Not supported - refer to parameter description'.
- Translate BXF Primary-BreakHeader event to:** A dropdown menu.
- Translate BXF Comment event to:** A dropdown menu showing 'Not supported - refer to parameter description'.
- Translate BXF Macro event to:** A dropdown menu showing 'Not supported - refer to parameter description'.
- Translate NonPrimary event to:** A section with an 'Ignore NonPrimary events that are not mapped' checkbox and a table.

The table has three columns: NonPrimaryEventName, Morpheus Event Type, and Special Action. Below the table are 'Add', 'Replace', and 'Delete' buttons. To the right of the table is a 'None' dropdown menu. At the bottom right is an 'Export NonPrimary event' button.

Fig. 2-59: BXF Schedule Service Configuration - Channels Event Type Configuration Tab

### - Translate BXF Primary Event To

No default.

Click on the channel(s) that this event type applies to, and from the drop-down menu select the Morpheus event type that the BXF Primary Event type will translate to.

#### **- Translate BXF Primary-ProgramHeader Event To**

No default.

This BXF event type is not currently supported by the BXF Schedule Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

#### **- Translate BXF Primary-BreakHeader Event To**

No default.

Click on the channel(s) that this event type applies to, and from the drop-down menu select the Morpheus event type that the BXF Primary-BreakHeader Event type will translate to.

#### **- Translate BXF Comment Event To**

No default.

This BXF event type is not currently supported by the BXF Schedule Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

#### **- Translate BXF Macro Event To**

No default.

This BXF event type is not currently supported by the BXF Schedule Service.

---

Note: A BXF event of this type must still be translated, otherwise the message will fail. It is therefore recommended that it is translated to a Morpheus event type that is disabled, i.e. has the 'enabled' flag set to false. Create an event dedicated to this function in the 'Configurator', and set the flag on the 'All Parameter Definitions' tab.

Disabled events will appear in the Editor.

---

#### - **Accept Events in the Past**

Default: disabled.

When un-ticked, a BXF message is rejected (i.e a NAK response is returned to the source system) if it contains a schedule with an event that is wholly in the past.

If ticked, all BXF messages are accepted regardless of the scheduling of the events contained therein.

---

Note: Apply on a per channel basis - select the required channel from the list prior to ticking the checkbox.

---

This function relates directly to the parameter **Margin for the On Air Event** in BXF Schedule Configuration - Common Parameters

## - Translate NonPrimary Event To

No defaults.

Non-primary event names (e.g. sub-events for GPIs and subtitles) are not specified in the BXF standard, and as such they are user definable.

---

Note: Any NonPrimaryEventName configured for this feature must match with that configured on the system from which messages will be received and responses transmitted.

---

For each channel that is destined to receive a user defined non-primary event in a BXF message, add entries as follows (Figure 2-60):

Translate NonPrimary event to:

☐ Ignore NonPrimary events that are not mapped

NonPrimaryEventName	Morpheus Event Type	Special Action

A B None C

Add Replace Delete Export NonPrimary event

Fig. 2-60: Translate NonPrimary Event To

- **Add**

- Click on a channel in order to select it.
- In the text box 'A', enter a name for the non-primary event to match that which is to be received in a BXF message.

---

Note: Names are case sensitive.

---

- From the drop-down menu 'B', select the Morpheus event type that the received non-primary event will be translated to.
- From the drop-down menu 'C', select any additional action that is to be applied in the translation (refer to 'Special Action' below).
- Click on the **Add** button - the new translation entry is added to the table.

- **Replace**

To modify an existing translation entry, click on any part of it and the text box and drop-down menus will be populated with its current properties. Enter alternative properties then click on the **Replace** button.

- **Delete**

To remove a translation entry from the table, click on any part of it and click on the **Delete** button.

## - Special Action

Additional actions that can be applied when an event is translated.

### AggregateIdenticalICEEvents

An aggregation feature whereby repetitions of an ICE sub-event are removed, retaining one that will be looped as many times as there were original sub-events, for the same duration and at the same frequency.

---

Note: Applies to ICE events only.

---

This is a tuning feature to preserve EventStore performance by preventing it from being overburdened.

It is suited, for example, to the repeated insertion of the same bug on the screen for very short durations.

Aggregation takes place according to the following rules:

- The sub-event types must be identical
- The sub-events must be of identical duration
- The time in between each sub-event in the sequence must be identical
- The sub-events must be consecutive
- The sub-events must all start within the same primary event

Several aggregations will be created if, for example, there are changes to the event duration or frequency over a number of repetitions.

### Example

**B5** = bug played for a duration of 5 seconds.

20 / 30 = duration in seconds between each bug.

**B5...20...B5...20...B5...20...B5.....30.....B5.....30.....B5...20...B5...20...B5...20**

Aggregation #1      Aggregation #2      Aggregation #3

In this case three aggregations are created due to the change in frequency of the bug payout.



### - Ignore non-Primary Events that are not Mapped

Default: disabled.

Ignore any non-primary events in a received BXF message that do not match an entry in the 'Translate NonPrimary Event To' table.

---

Note: If this function is disabled, the entire schedule will be rejected if it contains a non-primary event that does not match one of the configured translations, or one that exists in the EventStore.

---

### - Export NonPrimary Event

Save, to a file, the non-primary event translations configured for a selected channel. This function is only provided for administrative records - there is no import capability to allow sharing of the configured translations.

- a) From the channel list, select the one for which to export the translations.
- b) Click on the **Export NonPrimary Event** button.
- c) Windows Explorer is opened: enter a filename and destination folder, then click on the **Save** button.

---

Note: Only the non-primary events for the currently selected channel are exported.

---

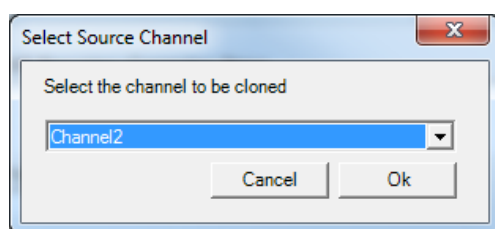
### - Clone Configuration From

Apply the set of existing primary and non-primary event translations on one channel to another:

1. Select the channel on which to apply the set of translations.
2. Click on the **Clone Configuration From** button.
3. The 'Select Source Channel' window is displayed (Figure 2-61).

From the drop-down menu, select the channel from which the translations will be cloned, then click on **Ok**.

The translation configuration is applied to the selected channel.



*Fig. 2-61: Select Source Channel*

---

Note: The receiving channel must support the event type related to the translation that is being cloned.

---

---

Note: This feature identifies whether or not the source event translations include the source channel name, and if so, that portion is changed to the name of the target channel during the cloning process.

---

## Messages Special Actions Tab

Service-wide parameters for this instance of the service.

Fig. 2-62: BXF Schedule Service - Messages Special Actions Tab

### - Schedule and Event Id Unique by Channel

Default: un-ticked.

By default, all schedules and events have unique GUIDs.

Check the box to allow different channels to load the same schedules and events with identical GUIDs.

---

Note: The title of this feature describes the default behaviour when disabled (un-ticked). If ticked then schedule and event ids can be duplicated across channels!

---

### - Find Previous Schedule By

The BXF Schedule Service finds the previous schedule based on information in either the Schedule Information Object (SIO) *BxfScheduleStartTime* or from the *InsertAfterEventId* element. It ensures that schedules are placed in the correct order as originally specified in the received BXF messages.

If schedules are cut & pasted to a different location, the original order (as specified by the start times in the BXF messages) cannot be used to find a previous schedule. This feature allows searches to be based on the actual start times of each schedule in a channel.

- **OriginalScheduleMessageDateTime**

The default setting.

The BXF Schedule Service finds the previous schedule based on information in the Schedule Information Object (SIO) *BxfScheduleStartTime*.

- **EventStoreScheduleStartDateTime**

The BXF Schedule Service finds the previous schedule based on the time at which it actually started (as shown in the EventStore), and not by its previously scheduled start time as shown against *BxfScheduleStartTime*.

### - SIO BxfScheduleStart Copy Name

No default.

Create an SIO property that, for each schedule received in the BXF message, the BXF Schedule Service will populate with the value of the schedule's start time. The start time of each schedule is that received in the message, and it will never be updated thereafter.

The purpose of this feature is to allow other applications to search for schedules, received in a BXF message, against their start time.

Enter a text string of choice as a name for the property. Once created, this property is visible in the Editor once the schedule is loaded.

### - Update Message Does Not Move Schedule

Default: disabled.

Check the box to prevent a BXF 'Update' message from changing the start time of a schedule (i.e. the time change is ignored). This option does not stop events in the schedule from being rearranged as prescribed in the message.

### - Update Message if do not Find Previous Event Insert on the End

Default: disabled. Tick the checkbox in order to enable this feature.

Applies only to a BXF message containing an 'update' action.

The default behaviour is such that, upon receipt of a BXF update message containing 1 or more events, the location to insert the update in the schedule is determined by a reference event in the message, i.e. the GUID of the event after which the updates should be positioned. If the specified reference event cannot be identified in the schedule, then the message is rejected.

In the case of a message containing updates for multiple schedules, if at any point a reference event cannot be identified, the remainder of the message is rejected. Such a situation could leave multiple schedules in an unexpected state with regards to BXF messaging, and may therefore require manual intervention to perform remedial actions.

When enabled, this function acts as a countermeasure to the reference event failing to be identified by positioning update events at the end of the reference event's schedule.

---

Note: This feature is related to 'Find Previous Schedule By', detailed above.

---

### Add Message Behaviour Pane

#### - Preprocess Action For Add Messages

The following options are available:

- **None** - default.

Use the standard BXF protocol for handling 'add' messages. This option disables the 'Process the Newest Message Only' option.

---

WARNING: This parameter should not be changed from the default if anything other than whole schedules are contained in BXF messages.

---

- **RemoveScheduleBeforeAdd**

Before processing an 'add' message, and if there are no locked events, then the existing schedule is deleted and replaced with the new schedule received in the 'add' message. This parameter can be used in conjunction with the **Ignore locked events** parameter below.

When this option is set, the schedule change is visible in the Editor.

- **Applyonlynecessaryforaddmessage**

The Morpheus schedule is synchronised to the schedule in the received BXF message, i.e. only the differences in the received message compared to the Morpheus schedule are applied. This parameter can be used with the **Ignore locked events** parameter below.

---

WARNING: This function must not be enabled unless BXF messages always contain whole schedules, otherwise messages that should be processed will be ignored.

---

#### **- Preset Event Close to Air Safety Limit**

Default: 10 seconds.

An amount of time that is added to the current EventStore time in order to create a window within which the Morpheus system is evaluated on its ability to process and ultimately play out a preset event identified in a received BXF message.

Any event received in a BXF message will be considered to be in the past (and will not therefore be played out) if its start time is within or inside the window:

$\text{<EventStore time>} + \text{<Preset Event Close to Air Safety Limit>}$

#### **Example**

With the default value of 10 seconds, if the EventStore time is 10:00:00:00 and the start time of the preset event received in the message is between 10:00:00:00 and 10:00:10:00, then it will be rejected.

#### **- Ignore Locked Events (if the schedule is full locked the message will be NAK)**

Default: disabled.

The default action is for events to be updated from a received BXF message only if they are all outside of the BXF Lock window. If any event is inside the BXF Lock window then the entire message is rejected.

When enabled, this feature allows updates to be applied only to those events that are outside of the BXF Lock window - any updates for events that are inside the BXF Lock window are ignored. However, if all of the events to be updated are inside the BXF Lock window, the whole message is rejected.

---

Note: This feature is related to the 'BXF Lock' window. Refer to BXF Lock Service.

---

#### **- Ignore Close to Air Safety Limit but do not Load Events on the Past**

Default: disabled.

This option only applies to top-level events.

When enabled, any time limit set in the 'Close to Air Safety Limit' option is ignored, although top-level events (and their children) in the schedule message, that do not occur in the past, are processed.

---

Note: In an enabled state, if, for example, a break header contains a commercial that occurs in the past and another commercial that occurs in the future, the whole break is rejected

---

**- Process the Newest Message Only (also will process the first removed message found)**

Default: disabled.

This function applies to BXF systems that are configured to only send complete schedules in a message.

If enabled, when a number of messages with identical SIOs are queued for processing, all but the latest message are ignored.

---

WARNING: This function must not be enabled unless BXF messages always contain whole schedules, as the outcome will be unpredictable.

---

'Accept Update and Delete on a Different Channel' Pane

**- Accept Update and Delete on a Different Channel**

Default: disabled.

If a BXF 'Update' or 'Remove' message is received for a schedule that has been moved to a channel that is different to the one indicated in the message (i.e. the channel indicated in the message does not match the channel in the EventStore for a particular schedule), enabling this feature forces the message to be accepted and processed for the schedule located on the channel specified in the EventStore (the channel specified in the BXF message is ignored).

Tick the checkbox to enable this feature.

---

Note: If the schedule is duplicated across two different channels, then the received BXF 'Update' or 'Remove' message will be responded to with a NAK., and the Morpheus schedule will not be modified.

---

**- Update NonPrimary Events Start Time with Primary Event Start Time if it is Zero**

Default: disabled.

Enable this feature if it is desirable to set non-primary events, that have been received in a BXF message with a start time of 00:00:00:00, to the start time of the parent primary event. When enabled, the start time of the parent primary event will be applied to all non-primary events in the BXF message with a start time of 00:00:00:00.

---

Note: This feature was created for a very specific customer requirement.

---

**- Set Event Depth Based on EventType (Set the Break Parent Based on Event Type)**

Default: enabled.

This function concerns associating break headers with primary commercials in the absence of the following elements in the BXF message:

- `<InsertAfterEventId>`
- `<PrimaryEventId>`

When enabled this function derives the association from the sequence in which the event types appear in the message

---

Note: Applies to BXF systems that are configured to only send complete schedules in a message.

---

---

WARNING: This function must not be enabled unless BXF messages always contain whole schedules, otherwise messages that should be processed will be ignored.

---

---

Note: This function was created for a very specific customer requirement.

---



## Offline Editor Mode & Export.sch Tab

Service-wide parameters for this instance of the service.

This tab uniquely defines parameters to allow a Morpheus schedule file (.sch) to be created from received BXF 'add' messages; any other message type is rejected.

Channels event type configuration Messages special actions **Offline editor mode & Export .sch** Other translation configuration Warning report c

**Offline Editor Mode**

The Offline Editor Mode is a specific operational mode to only mark BXF schedule messages as processed but does not update Eventstore. It can be used together with Export in order to only generate Morpheus .SCH schedule files.

☐ Offline Editor Mode

**Export .sch**

The Export .sch option is a specific operational mode to generate Morpheus .SCH schedule files. The file generated can then be used by the Offline Editor.

There are restrictions on the capabilities of this feature and how it works:

- It uses the online Eventstore, which implies that the local current\_system.xml file used by the Offline Editor must always be kept in sync with the live Eventstore.
- It is designed to process only BXF schedule messages whose action='add'. BXF schedule messages whose action='update' or 'remove' will be ignored.
- If the exported .SCH schedule file already exists, this mode will create a new version of the file appending [n] to its filename.
- This mode does not manage the folder where the exported .SCH schedules files are saved.

☐ Export .sch files ☐ Replace non existing Uid references by -1

File Name

- ☐ First Primary Program Name
- ☒ First Primary Material Id
- ☒ Schedule Id
- ☒ Message Id
- ☐ Channel name
- ☐ Start datetime

Move Up Move Down

Path to save the .SCH files

File name example

[FIRSTMAT][51f043ae-7f35-4f48-ac2e-719851c838df][19c687a3-703b-4a9e-a698-d0784411a68e].SCH

Fig. 2-63: BXF Schedule Service - Offline Editor Mode & Export.sch Tab

### - Offline Editor Mode


Default: disabled.

When enabled (ticked), BXF schedule messages will be processed and validated but the local EventStore will not be updated. This option can be used with the 'Export .sch' function for the sole purpose of generating schedule (.sch) files.

### - Export .sch / Path to Save the .SCH Files / File Name and Move Up / Move Down

Default: disabled.

When enabled, for each 'add' message received, a schedule file will be generated for import into an Offline Editor.

1. Select a folder in which to store the .sch files using the **Path to Save the .SCH Files** ellipsis button .
2. Select a filename format from one or more of the options in 'File Name' menu, as follows:
  - **First Primary Program Name**  
Filename format: *[FirstProgrammeName].sch*
  - **First Primary Material Id**  
Filename format: *[FirstMat].sch*
  - **Schedule Id**  
Filename format: *[6aff294c-7fa0-4b17-ab05-20939c5a608b].sch*
  - **Message Id**  
Filename format: *[19c687a3-703b-4a9e-a698-d0784411a98e].sch*
  - **Channel name**  
Filename format: *[Channel1].sch*
  - **Start datetime**  
Filename format: *[2014-05018T13:00:00Z].sch*
  - **First Comment Title**  
The service identifies the first eventType attribute "Comment" in the BXF message, retrieves the title text from the <EventTitle> element and adds it to the filename

#### Example

```
<EventData eventType="Comment">  
  <EventTitle>Columbo</EventTitle>  
</EventData>
```

Create a unique structure for the filename by ticking multiple options that will be combined, the order of which can be set by selecting each format type and applying the **Move Up** and **Move Down** buttons.

A representation of the selected structure of the filename is displayed dynamically along the bottom of the window, as the checkboxes are ticked.

---

Note: At least one file name format must be checked, otherwise an error will be displayed.

---

The following operational restrictions apply to this feature:

- This function uses the online EventStore, therefore the `current_local.xml` file used by the Offline Editor must always be kept in sync with the live EventStore
- It is designed to process only BXF schedule messages whose action = 'add'. BXF schedule messages whose action = 'update' or 'remove' will be ignored
- If the exported `.sch` schedule file already exists, this mode will create a new version of the file appending `[n]` to its filename
- This mode does not manage the folder where the exported `.sch` schedules are saved

### - Replace Non Existing Uid References by -1

Default: disabled.

This function concerns generating a schedule file (`.sch`) from a received BXF message in which there are events that are partially or wholly in past.

When enabled, a message of type 'add' is processed as follows:

1. Assign the number -1 to the *PreviousUid* and *OwnerUid* properties of the first event, and allocate a consecutive number to each event thereafter in order to create an ordered sequence.

---

Note: The number -1 indicates that there are no previous events.

---

2. Remove all events that are partially or wholly in the past.

The removal of any events breaks the numbering sequence and renders the schedule incomplete - the Morpheus system will only allow a complete schedule to be loaded in the Editor.

3. Inspect the *PreviousUid* and *OwnerUid* properties of all events that reference other events that no longer exist (due to stage 2 above), then allocate the number -1 to each.

---

Note: This function was created for a very specific customer requirement.

---

## Other Translation Configuration Tab

Service-wide parameters for this instance of the service.

### - TransitionRate

The BXF standard specifies three video / audio event transition rates: 'Fast', 'Medium', and 'Slow' - these are a representation of actual frame rates. Morpheus, however, processes actual frame rates only, and does not recognise the BXF names. In order for Morpheus to be compatible with the BXF standard, a translation mechanism is accessible from the 'Other Translations Configuration' tab.

Channels event type configuration Messages special actions Offline editor mode & Export .sch Other translation configuration Warning report c

**TransitionRate**

Transition rate is used the speed of a Video or Audio transition.  
In BXF exist 3 TransitionRate: Fast, Medium and Slow, Morpheus otherwise work with number of frames.

Fast 00:00:00:00

Medium 00:00:00:00

Slow 00:00:00:00

Replace the original duration based on repetition

Original duration	Repetition	New duration
-------------------	------------	--------------

Enter original duration Event count New duration

Add Replace Delete

Fig. 2-64: BXF Schedule Service - Other Translation Configuration Tab

The TransitionRate fields allow specific frame rates (that will be understood by Morpheus) to be entered against the BXF named transition rates. When a BXF message is received by this service, indicating a transition rate of 'Fast', 'Medium', or 'Slow', it is translated to the configured actual frame rate, which is then applied.

## - Replace the Original Duration Based on Repetition

A search & replace mechanism that adjusts the timecode of events in order to compensate for a traffic system not accounting for drop-frame when generating a schedule that is delivered to this service in a BXF update message.

Applies only to NTSC systems.

---

Note: This feature was developed for a specific customer requirement.

---

A search is performed on the received BXF message for all events that match a configured timecode (*Original Duration*). At a configured interval (*Repetition*), the timecode of each event is adjusted by the number of frames specified (*New Duration*), before forwarding the schedule to the EventStore.

---

Note: This feature will recognise both SMPTE Timecode and UTC Duration in the received BXF message, however the search & replace configuration must use SMPTE Timecode only.

---

---

Note: In order for the timecode change to be accurate, 2 frames must be removed in every 60 seconds from the received schedule.

---

- **Add**

Create an entry from adding the following information:

- a) Original Duration

The timecode of the Primary event upon which to base the search in the received BXF schedule update message. Specify using SMPTE Timecode only.

---

Note: Each filter must be configured with a unique 'Original Duration'.

---

- b) Repetition

The interval at which the timecode will be changed, e.g. if the value of 'Repetition' is configured as 3, then the timecode will be changed on every third event found in the BXF message.

- c) New Duration

The alternative timecode to apply to the events. Specify using SMPTE Timecode only.

---

Note: There is no limit to the number of 'search & replace' entries that can be created.

---

- **Replace**

Click on any part of an existing search & replace entry - the text entry boxes are populated with the existing values. Modify the values as required and click on **Replace**.

- **Delete**

Click on any part of an existing search & replace entry then click on **Delete**. This function deletes the whole entry.

### Example

In order to achieve the correct frame count per minute, for 4 events in the received BXF message, each having a timecode of 00:00:15:00, configure a search & replace entry as follows:

Original Duration: 00:00:15:00

Repetition: 2

New Duration: 00:00:14:29

---

Note: For information only.

UtcDuration in the received BXF message is specified in the following manner *PnYnMnDTnHnMnS*, where:

- *P* indicates the period (required)
  - *nY* indicates the number of years
  - *nM* indicates the number of months
  - *nD* indicates the number of days
  - *T* indicates the start of a time section (required when specifying hours, minutes, or seconds)
  - *nH* indicates the number of hours
  - *nM* indicates the number of minutes
  - *nS* indicates the number of seconds
-

## Warning Report Configuration Tab

Service-wide parameters for this instance of the service.

The configuration of 'Warning Reports' allow an operator to be provided with alerts in the Morpheus Editor for schedule updates, or attempted schedule updates, that have been processed from received BXF messages.

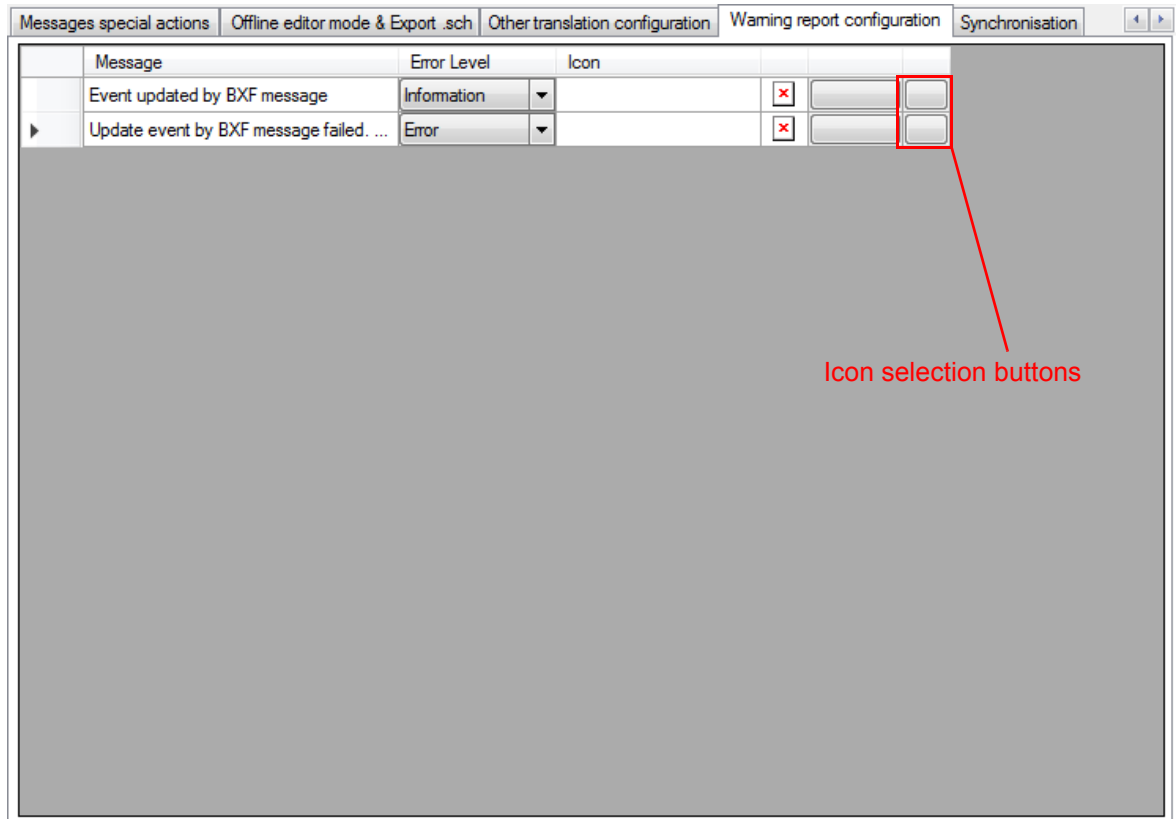


Fig. 2-65: BXF Schedule Service - Warning Report Configuration Tab

Two message types are defined:

- **Event updated by BXF message**

A BXF message has been received that has successfully made a change (update) to the original schedule.

- **Update event by BXF message failed. Event Locked/OnAir by Event**

A BXF message has been received that intended to make a change (update) to the original schedule, but the change has failed because either the event that was to be changed is locked in the schedule, or the event that was to be changed is already on air. In response to this condition, a NAK message is returned to the originating system.

For each message type defined above, one of the following alert 'Error Levels' (severity levels) can be set from the drop-down menu:

1. None
2. Information
3. Warning
4. Error
5. Critical

---

Note: A different 'Error Level' must be attributed to each of the message types, e.g. they cannot both be set to 'Informative'.

---

The 'Error Levels' are purely administrative, are designed only to be informative to an operator, and have no influence upon any process.

Additionally, an icon can be configured to appear in the 'Validation' column of the Morpheus Editor against the event that is related to one of the two received BXF message types detailed above.

1. Click on the right-most blank button in-line with the message (refer to Figure 2-65) - this will open a browser.
2. Select an appropriate icon and click on the **Open** button in the browser.

The link to the icon will appear in the 'Icon' column on the 'Warning Report Configuration' page, and the icon itself will appear alongside it.

---

Note: Double click in the 'Validation' column against the event to view the 'Event Status Report' that will contain one of the error messages defined above.

---



## Synchronisation Tab

Service-wide synchronisation parameters for this instance of the service, applicable to all synchronisation modes, i.e. N+1, Tertiary, and MultiSite Sync.

**BXF Schedule Configuration**

Service version: **V1**

BXF Repository Connection String: **Data Source=172.31.7.15;User ID=sa;Password=sa;Initial Catalog=BXF\_225\_0\_11\_116** Test connection

Processing Agent Name: **BXF Schedule** Close to Air Safety Limit: **0** minutes Margin for the on air event: **10** seconds Maximum number of subevents: **60**

BXF Schema Version for messages generated: **V1.0 2008** ☐ Do not load schedules containing Primary events with zero durations

Order to process the messages: **GeneratedDateTime** Language for attributes with language option: **eng**

☒ Store processed message and SCH schedule when possible

Path to store the processed message: **E:\BXFScheduleMessagesLog\Complete**

Path to store the processed message with errors: **E:\BXFScheduleMessagesLog\Error**

**Messages special actions** | Offline editor mode & Export .sch | Other translation configuration | Warning report configuration | **Synchronisation**

Time out between status: **60** seconds

Time to retry synchronize the channels: **300** seconds

System name	BXF Schedule change notifier Agent Name	BXF Query Agent Name

**Add** **Replace** **Delete**

Channels to not automatic resync when fail

ChannelName

**Add** **Replace** **Delete**

**Cancel** **Save**

Fig. 2-66: BXF Schedule Service - Synchronisation Tab

### - Time Out Between Status

The amount of time that this service will wait for a response to a request before declaring the sync partner as unavailable.

The default is 60 seconds, which is the highest recommended value. It should be reduced according to the performance of the network over which the systems are synchronised - the higher the performance, the lower the value.

### - Time To Retry Synchronise The Channels

The amount of time after **Time Out Between Status** has expired that this service will resend an unanswered request.

The default is 300 seconds. The recommended value is 60 seconds.

---

Note: This service will constantly retry at the specified interval should a response not be received.

---

### - Add / Replace / Delete

The properties added are correlated with those received from the EventStore in order to establish communication of BXF services between a Source and Target sync pair.

- **Add**

- a) Enter the system name of any potential sync partner, as defined in their EventStore configurations.

---

Note: Parameters are case sensitive.

---

- b) Enter the BXF Schedule Change Notifier agent name of the service configured on the sync partner.
- c) Enter the BXF Query agent name of the service configured on the sync partner.
- d) Click on **Add**. The names are added to the main pane.

---

Note: Warning messages are displayed if any system / agent name fields are blank, or if the system name is already configured for this instance of the service.

---

- **Replace**

Highlight the system / agent entry to be changed - the text entry boxes are populated with the existing entries. Enter the new system name and / or agent name in the appropriate text box and click on **Replace**.

- **Delete**

Highlight the system / agent entry to delete then click on **Delete**. This function deletes the whole system / agent entry.

---

Note: For X & Y systems in Multi-Site Sync environments, an individual service agent must be created for local System X and local System Y, each of which must be configured with an entry for System X and System Y of the remote sync partner.

---

## - Channels to not Automatic Resync When Fail

---

Note: This feature applies to Multi-Site Sync environments only

---

On the Target system only, specify those channels that should not attempt to automatically resynchronise following a loss of connection.

- **Add**

- a) From the drop-down menu, populated from the EventStore, click on a channel.
- b) Click on **Add**. The selected channel appears in the 'ChannelName' pane.

- **Replace**

Click on the channel name to be changed. From the drop-down menu, select an alternative channel and click on **Replace**.

- **Delete**

Click on the channel name to delete then click on **Delete**.

---

Note: Whilst a channel is configured to not resynchronise, an automatic retry will only occur if the channel is restarted using the Multi-Site Control User Interface.

---

## BXF SOAP Reader Service

---

Note: This service has been deprecated. Refer to software engineering with any queries.

---

This service receives a BXF message through a SOAP connection and validates it according to the BXF schema - if valid, it is stored in the BXF database awaiting processing.

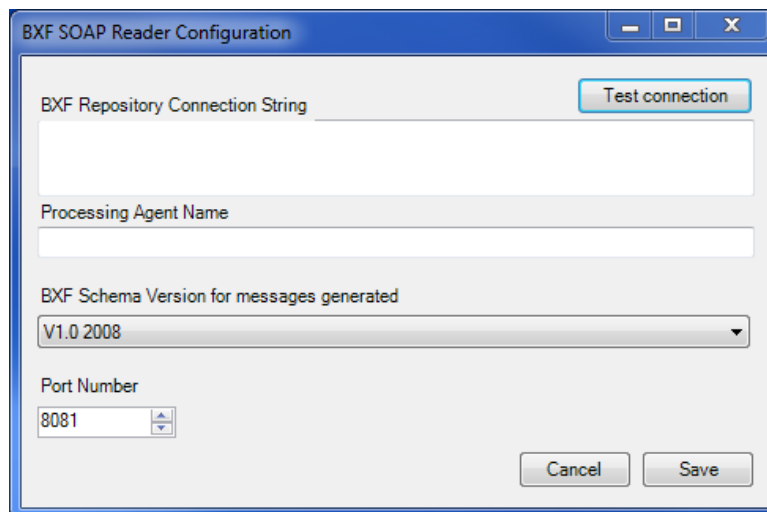


Fig. 2-67: BXF SOAP Reader Service Configuration Window

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxfl catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

#### **- Processing Agent Name**

Enter a name that will identify the BXF SOAP Reader Service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). Exceptionally, the Agent Name attributed to a BXF SOAP Reader service is only used to record its identity in the BXF database as the agent responsible for depositing a particular message. The Agent Name of a BXF SOAP Reader service is never used as addressing information by any other service.

---

Note: It is recommended that the agent name is different to the instance name.

---

#### **- BXF Schema Version for Messages Generated**

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

#### **- Port Number**

Select an IP protocol port number from which to receive the SOAP messages.

## BXF SOAP Writer Service

Delivers BXF messages using SOAP through a WEB Service - BXF messages are not created.

---

Note: This service has been deprecated. Refer to software engineering with any queries.

---

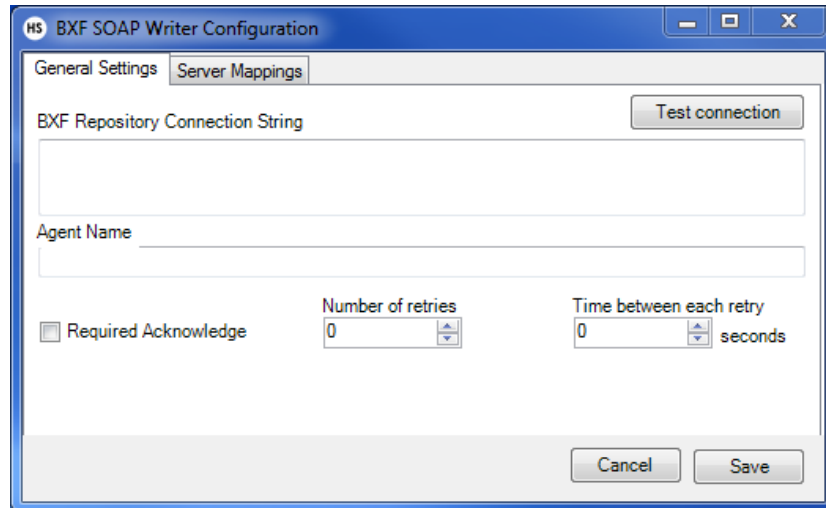


Fig. 2-68: BXF SOAP Writer Service Configuration Window - General Settings Tab

## General Settings Tab

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxm catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalog=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Agent Name

Enter a name that will identify the service - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). Exceptionally, the Agent Name attributed to a BXM SOAP Writer service is only used to record its identity in the BXM database as the agent responsible for retrieving a particular message. The Agent Name of a BXM SOAP Writer service is never used as addressing information by any other service.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - Required Acknowledge

Tick the checkbox if, for each message sent, the system must wait for an acknowledgement to be returned before sending the next message to the same destination. If no acknowledgement is received for the value of **Time Between Each Retry**, then the same message will be sent again and for as many times as defined for the value of **Number Of Retries**.

## Server Mappings Tab

Define the BXF destinations with which this service will communicate.

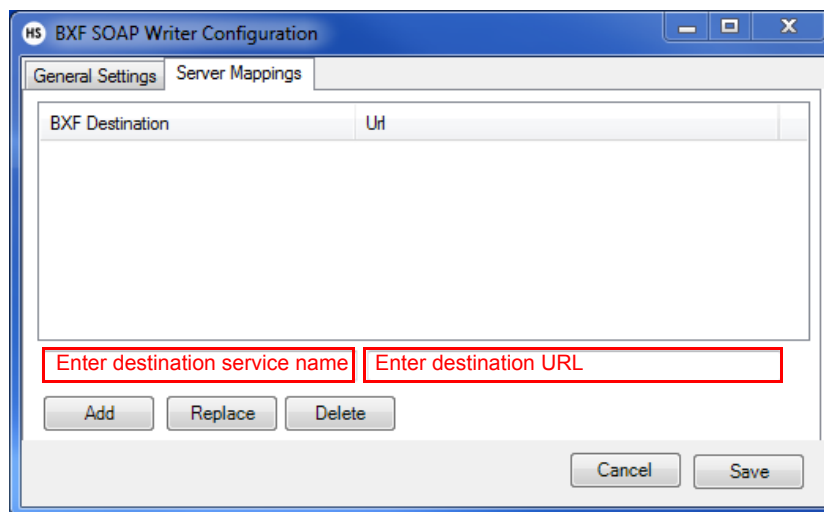


Fig. 2-69: BXF SOAP Writer Service Configuration Window - Server Mappings Tab

### - Add / Replace / Delete

Refer to Figure 2-69.

- **Add**

- a) Enter the name of the destination service to which messages are to be sent.

---

Note: Parameters are case sensitive.

---

- b) Enter the URL of the destination service.

- c) Click on **Add**. The destination service and its URL will appear in the main pane.

- **Replace**

Highlight the destination entry to be changed - the text entry boxes will be populated with the entry. Enter the new destination properties in the appropriate text boxes and click on **Replace**.

- **Delete**

Highlight the destination entry to delete then click on **Delete**. This function deletes the whole destination entry.

---

Note: By adding multiple destinations, a single SOAP Writer Service is able to write to many SOAP Reader Services.

---



## BXF Socket Reader Service

The BXF Socket Reader Service reads BXF messages that have been transmitted from one or more BXF Socket Writer services (or equivalent services from a third party external system).

Messages are read from the configured TCP Socket and validated against the BXF schema:

- If the message is valid it is deposited into the local BXF database
- If the message is invalid, then it is ignored (as the source system address is not read from a failed message, it is not possible to return a NAK)

### BXF Socket Reader Service: Principle of Operation

The functionality of the BXF Socket Reader service is shown in Figure 2-70 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

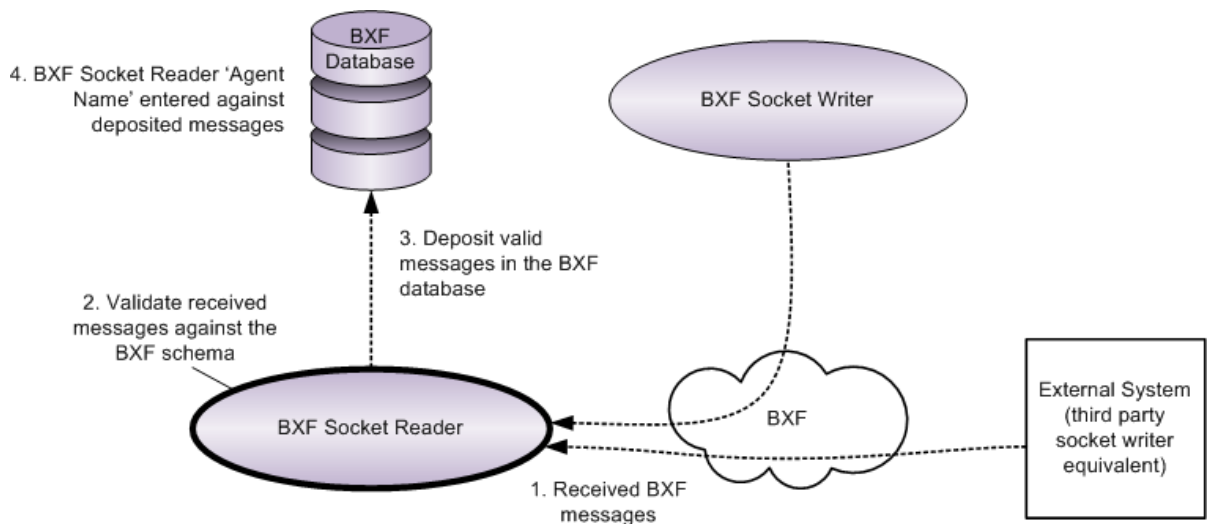


Fig. 2-70: BXF Socket Reader Service Logical Topology

## BXF Socket Reader Service Configuration

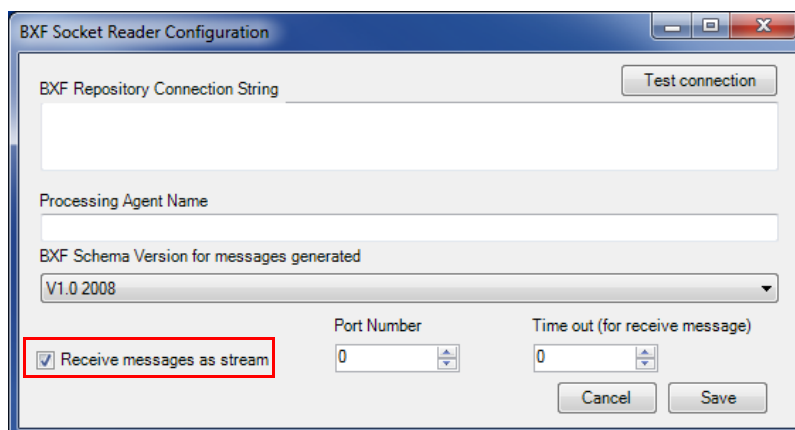


Fig. 2-71: Configuring the BXF Socket Reader Service

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxr catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Processing Agent Name

Enter a name that will identify the service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). Exceptionally, the Agent Name attributed to a BXF Socket Reader service is only used to record its identity in the BXF database as the agent responsible for depositing a particular message. The Agent Name of a BXF Socket Reader service is never used as addressing information by any other service.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - BXF Schema Version for Messages Generated

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### - Receive Message As Stream

When ticked, this service will process all messages in the order in which they were received, before depositing them into the local BXF database. This parameter must be ticked in order to support any of the BXF Synchronisation modes.

---

Note: The associated BXF Socket Writer Service must have **Send Message As Stream** enabled.

---

---

WARNING: This parameter must not be used for any purpose other than for the BXF Synchronisation modes.

---

### - Port Number

The IP port number on which this agent will listen for messages from a BXF Socket Writer.

---

Note: The port number must match that which is configured on a BXF Socket Writer service.

---

---

Note: This service is addressed using the IP address of the host device on which it is installed.

---

### - Timeout (for Receive Message)

The amount of time in seconds after which this service will close the connection, having received no messages from any BXF Socket Writer. This parameter is only valid when using the default message transfer mode which opens and closes a connection for each message, i.e. messages are not streamed.

The default is 0 seconds.

The recommended value is 10 seconds.

---

Note: This parameter is disabled if **Receive Message As Stream** is enabled.

---

## BXF Socket Writer Service

The BXF Socket Writer service queries the BXF database for messages with destination services that it is configured for, transmitting them via an IP socket.

---

Note: The BXF Socket Writer only sends messages to the services configured as destinations.

---

### BXF Socket Writer Service: Principle of Operation

The functionality of the BXF Socket Writer service is shown in Figure 2-72 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

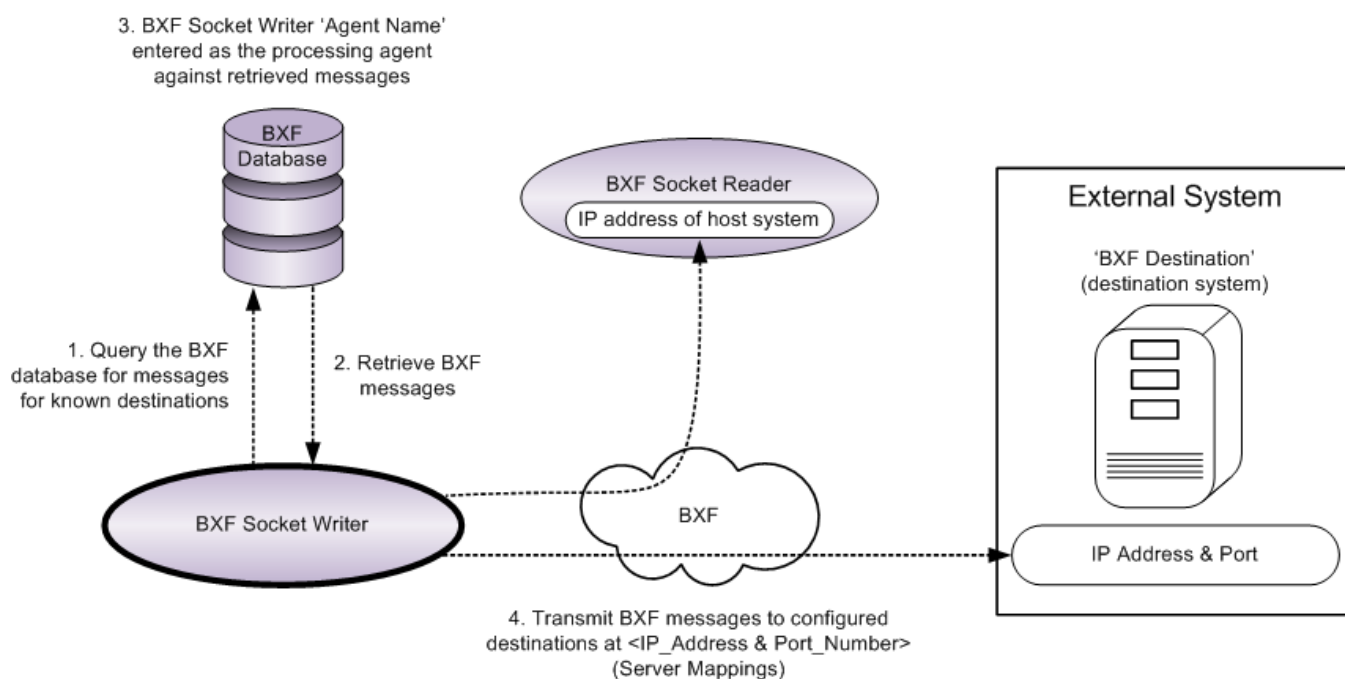


Fig. 2-72: BXF Socket Writer Service Logical Topology

## General Settings Tab

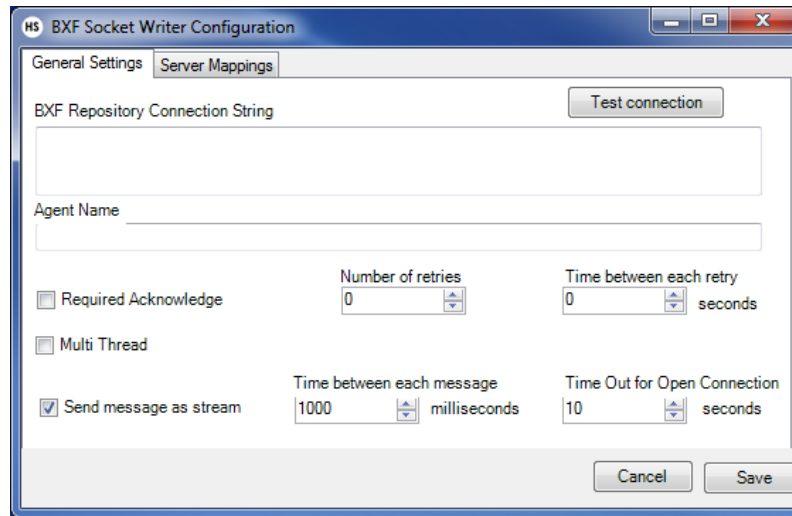


Fig. 2-73: Configuring the BXF Socket Writer Service: General Settings Tab

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxm catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### - Agent Name

Enter a name that will identify the service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). Exceptionally, the Agent Name attributed to a BXF Socket Writer service is only used to record its identity in the BXF database as a processing agent of a particular message. The Agent Name of a BXF Socket Writer service is never used as addressing information by any other service.

---

Note: It is recommended that the agent name is different to the instance name.

---

### - Required Acknowledge

Tick the checkbox if, for each message sent, the system must wait for an acknowledgement to be returned before sending the next message:

- To any destination when this service is not configured for 'Multi Thread'.
- To the same destination when this service is configured for 'Multi Thread'.

If no acknowledgement is received within the time attributed to the value of **Time Between Each Retry**, then the same message will be sent again and for as many times as defined by the value of **Number Of Retries**.

---

IMPORTANT: This feature must not be enabled for the Multi-Site Sync mode.

---

### - Number of Retries

The number of times that a single message will be resent to a destination before being marked as having failed. Messages are resent if no acknowledgement of receipt has been returned.

The default is 0.

The recommended value is 3.

---

Note: This parameter applies only if **Required Acknowledge** is enabled.

---

### - Time Between Each Retry

Related to 'Number of Retries'.

The number of seconds in between each attempt to send a message.

The default is 0 seconds.

The recommended value is 10 seconds.

#### - Multi Thread

Default: disabled (un-ticked).

When enabled, this service will send messages to configured destinations concurrently.

When disabled, this service will send messages to configured destinations consecutively, in the order in which they are listed in the 'Destination' table, and regardless of the number of destinations.

#### - Send Message As Stream

Default: disabled (un-ticked).

When enabled, a connection is maintained in an open state whenever possible.

---

**IMPORTANT:** This feature must be enabled in order to support Multi-Site Sync.

---

---

**WARNING:** This parameter is specific to all modes of BXF Synchronisation and must not be used for any other purpose.

---

---

**Note:** The associated BXF Socket Reader Service must have **Receive Message As Stream** enabled.

---

#### - Time Between Each Message

Related to **Send Message As Stream**.

The time, in milliseconds, between each streamed message.

Default: 1000 milliseconds (recommended).

#### - Timeout for Open Connection

Default: 10 seconds (recommended).

For each connection that is opened, regardless of whether or not streaming is enabled, the service will wait for the configured amount of time before closing the connection should there be no reply from the destination.

This feature is hardcoded to retry a maximum of 3 times for each new connection that is opened.

---

**Note:** When streaming is enabled, a single connection remains open for each configured destination.

When streaming is disabled, a connection is opened for a single message transmission and then closed.

---

## Server Mappings Tab

Define the destination services to which this BMF Socket writer will transmit messages.

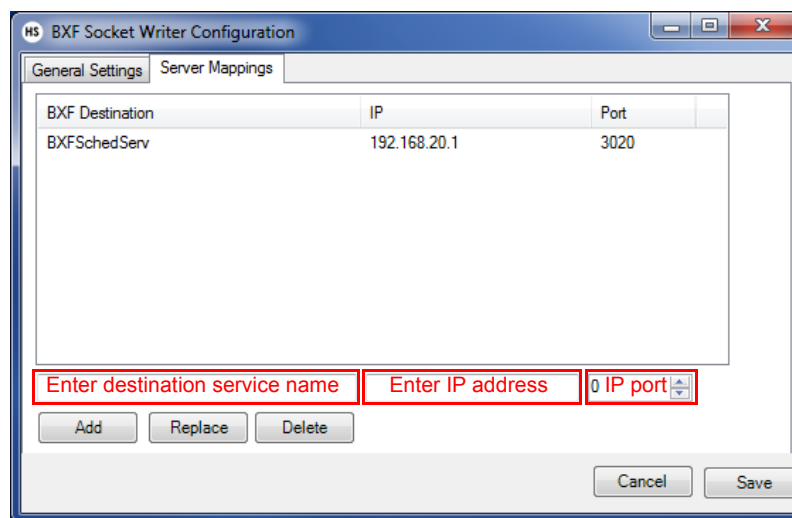


Fig. 2-74: Configuring the BMF Socket Writer Service: Server Mappings Tab

### - Add / Replace / Delete

Refer to Figure 2-74.

- **Add**

- a) Enter the name of the destination service to which messages are to be sent.

---

Note: Parameters are case sensitive.

---

- b) Enter the IP address of the destination service to which messages are to be sent.
- c) Enter the IP port number on which the destination service has been configured to listen.
- d) Click on **Add**. The BMF destination appears in the Server Mappings pane.

- **Replace**

Highlight the destination entry to be changed - the text entry boxes will be populated with the entry. Enter the new destination properties in the appropriate text boxes and click on **Replace**.

- **Delete**

Highlight the destination entry to delete then click on **Delete**. This function deletes the whole destination entry.

---

Note: By adding multiple destinations, a single Socket Writer Service is able to write to many Socket Reader Services.

---



## MAPP BXF Transfer Request Service

A MAPP agent that retrieves messages from the MAPP database that are addressed to it, and performs the following tasks:

- Generates 'BXF Content Transfer Requests' for material that is unavailable
- Processes BXF Query messages, from an external system, for all 'BXF Content Transfer Requests' that have been acknowledged as received but not acknowledged as completed. This is implemented in order to allow external systems to retrieve unprocessed messages that have been erased locally, perhaps due to a reboot
- Processes material transfer request updates from an external system

---

Note: If this service retrieves multiple Transfer Requests from the MAPP database for the same Material Id, it will integrate them into a single 'BXF Content Transfer Request' message for delivery to the destination external system.

---



---

Note: This service posts instructions for other services, such as VNM, to copy or move files (it has no responsibility to perform these actions itself).

---

The following BXF standard 'Transfer Types', contained within a message of type 'Information', can be processed by this service:

- **Duplication**

Definition: create a new entry, regardless of what may or may not already exist on the MAPP database relating to the information contained within the message.

This implementation of BXF only supports the receipt of this message type from an external system - this message type will never be generated for transmission to an external system.

- **File Transfer**

Definition: transfer a material file from one location to another then register the new location in the MAPP database.

- **Content Transfer Update message**

Definition: a message type that permits a third party system to send content transfer update messages to a requesting system in order to show the progress of a content transfer.

Refer to Content Transfer Update Message.

---

Note: All messages relating to a specific task have identical TransferId's until the transaction is either completed or failed.

---

The following MAPP transfer types, contained within a BXF message of type 'Information', can be processed by this service:

---

Note: Although these transfer types do not adhere to the BXF standard, their meaning is interpreted from the entry in the <Source> element in a BXF message, in the manner described below.

---

- **Purge Request / Delete Request**

Definition: delete the specified Material Id on the specified device then remove its entry from the MAPP database.

This transfer type is used for database storage space administration.

This transfer type is implied if a destination is specified in the <Source> element of the BXF message.

- **Kill Request**

Definition: delete all instances of the specified Material Id from all devices then remove all associated entries from the MAPP database. This transfer type can be used, for example, to remove all material upon the expiry of its broadcast licence.

This transfer type is implied if no destination is specified in the <Source> element of the BXF message.

## MAPP BXF Transfer Request Service: Principle of Operation

The functionality of the MAPP BXF Transfer Request service is shown in Figure 2-75 in relation to the other services upon which it relies in order to fulfill its role.

The dashed lines indicate the BXF message transmission path, the dash / dot lines indicate the implied reposting of a Transfer Request onto the MAPP database for the attention of another agent, and the solid lines represent the Morpheus / MAPP delivery mechanism.

Note: Although the example in Figure 2-75 shows the Transfer Decorator service as being that which originally posted the Transfer Request, the MAPP BXF Transfer Request Service will process messages deposited in the MAPP database that are addressed to it from any number of other services.

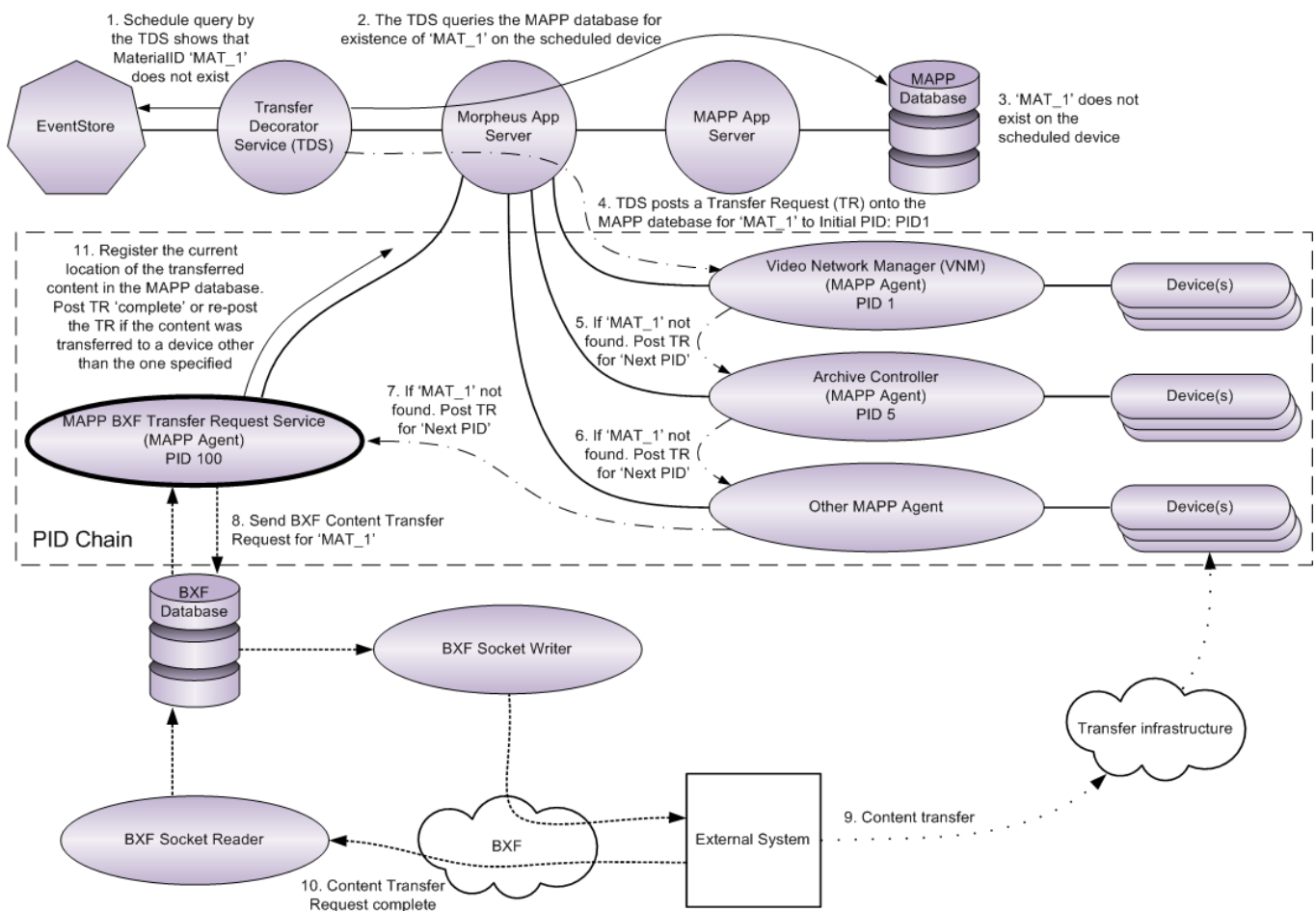


Fig. 2-75: MAPP BXF Transfer Request Service - Logical Topology

Note: A MAPP database query from the Transfer Decorator Service can be transmitted directly to the MAPP database from the Morpheus App Server, bypassing the MAPP App Server as shown in stage 2 of Figure 2-75.

---

Note: In stages 4, 5, 6, and 7, as a result of the current owner agent of the Transfer Request (TR) message failing to find the required material, it posts it back onto the MAPP database for the attention of another agent, identified by the 'Next Process ID' parameter. The sequential reposting of the message creates a PID Chain (for further information, refer to MAPP Process IDs (PIDs) and PID Chains).

---

---

Note: In stage 9, it is possible that the external system may transfer the content to a location other than the one specified in the TR, as the MAPP BXF Transfer Request service has no control over the process. In the event that the content is transferred to another server, as indicated in the Content Transfer Request Complete message received back from the external system (stage 10), then the MAPP BXF Transfer Request service will perform the following tasks (stage 11):

- Post a Transfer Request Complete message onto the MAPP database indicating the current location of the content
  - Post a TR onto the MAPP database requesting that the content be transferred to the original target server (refer to **Repost the Transfer when the Material is not on the Final Target**)
-

## Content Transfer Update Message

The BXF standard specifies a message type that permits a third party system to send content transfer update messages to a requesting system in order to show the progress of a content transfer. This has been adopted in the MAPP BXF Transfer Request Service, and requires no configuration. The progress of a content transfer is displayed as a percentage progress bar in the 'Main' column of the Editor, against the material related to the transfer.

To display the progress bar, in the Editor click on Tools > Configuration > Common and tick the **Show Material Transfer Progress** checkbox, as shown below in Figure 3. Optionally, change how frequently the progress bar is updated in seconds (the default, and minimum setting, is 10 seconds).

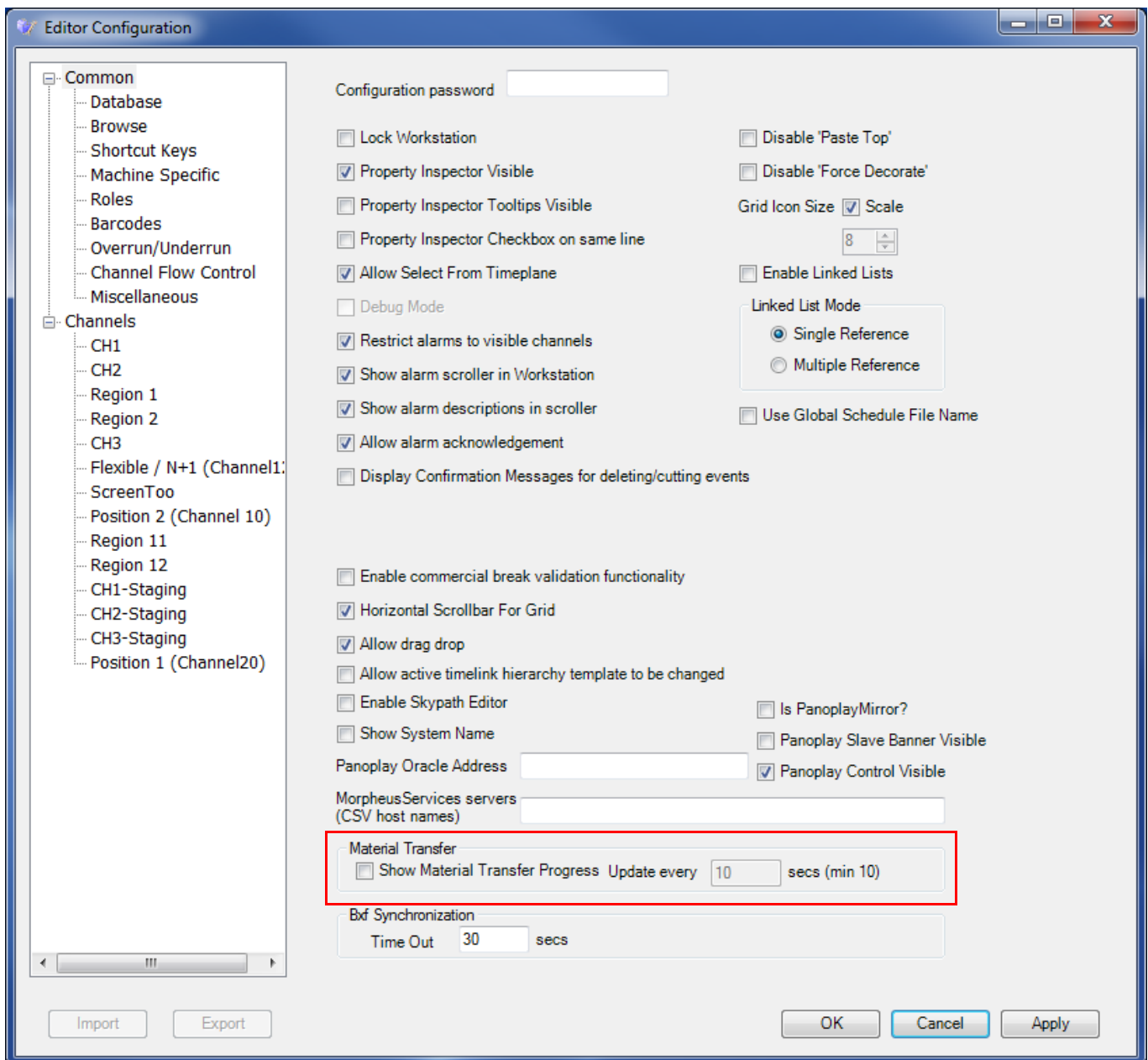


Figure 3 Enabling Material Transfer Progress in the Editor Configuration Window

## General Settings Tab

The screenshot shows a Windows-style configuration window titled "Mapp BXF Transfer Request Service Configuration". It has several tabs: "General Settings" (selected), "Transfer settings", "Owner and target device rules", "Extra information at BXF private section", and "Automatic". The "General Settings" tab contains the following fields and controls:

- BXF Repository Connection String:** A large text area for entering the connection string, with a "Test connection" button to its right.
- Agent Name:** A text input field.
- Destination:** A text input field.
- BXF Schema Version for messages generated:** A dropdown menu currently showing "V1.0 2008".
- Process ID:** A numeric spinner box set to 0.
- Next Process ID:** A numeric spinner box set to 0.
- Process ID to repost request to when material is available:** A numeric spinner box set to 0.
- Time between the process be executed:** A numeric spinner box set to 60, followed by the unit "seconds".
- Time Required Check Interval:** A numeric spinner box set to 60, followed by the unit "seconds".
- Process ID for new request (0 disable):** A numeric spinner box set to 0.
- Next Process ID for Delete/Kill Requests:** A numeric spinner box set to 0.

At the bottom right of the window are "Cancel" and "Save" buttons.

Fig. 2-1: MAPP BXF Transfer Request Service - General Settings Tab

### - BXF Repository Connection String

Enter the connection string that will provide access to the BXF database. Define the connection string as SQL Server SqlConnection value pairs separated by semicolons (;) in the following format:

Data Source=<host>;User ID=<user id>;Password=<pwd>;Initial Catalog=<bxg catalogue>

Where:

Data Source=<SQL server address [typically the machine name]>;

User ID=<SQL server login account [default = sa]>;

Password=<SQL server login account password [default = sa]>;

Initial Catalogue=<DatabaseName> (the name of the database with which to connect).

---

Note: The properties for this string will have been configured at the time of the database creation.

---

Verify the validity of the connection string by pressing the **Test Connection** button.

### **- Agent Name**

Enter a name that will identify the service as a processing agent - any name is permissible, as long as it is unique (except when configured for redundancy [specific services only]). The Agent Name is used as the identity of the originating system ('origin' field entry of the Messages table in the MAPP database) when sending a BXF message, and as the Destination System ('destination' field entry of the Messages table in the BXF database) for receiving messages.

---

Note: It is recommended that the agent name is different to the instance name.

---

### **- Destination**

A BXF 'Writer' service is responsible for delivering messages, that have been deposited into the BXF database by this service, to its destination.

Enter the name of the destination service to which a BXF 'Writer' service will deliver the message.

---

Note: If no 'Writer' service has been configured, messages will remain on the database until removed by the BXF Database Maintenance Service.

---

### **- BXF Schema Version for Messages Generated**

Specify the version of the BXF Schema against which this service will generate messages.

---

Note: It is recommended that all BXF service instances used in the deployment of a solution are configured with the same schema version.

---

---

Note: The BXF services support all of the schemas in order for received messages to be validated accordingly.

---

### - Process ID

The process ID of this instance of the 'MAPP BXF Transfer Request Service'. It will be used by other services to identify this instance as the destination service in a transfer request message.

It is recommended that the 'MAPP BXF Transfer Request Service' is configured with a PID of 100, and that other agents are configured with lower PIDs as suggested in Table 2-5. It is preferable to create PID chains based upon efficiency of performing the required task, and that this instance of the service is the last in the PID chain.

For information relating to PID chains, refer to MAPP Process IDs (PIDs) and PID Chains.

*Table: 2-5*

PID	Service
5	Video Network Manager (VNM)
10	Archive Controller
15	Other MAPP service
100	This service

---

Note: The Process ID (PID) property is specific to MAPP; it is not part of the BXF specification, and is not therefore included in any BXF messaging.

---

### - Next Process ID

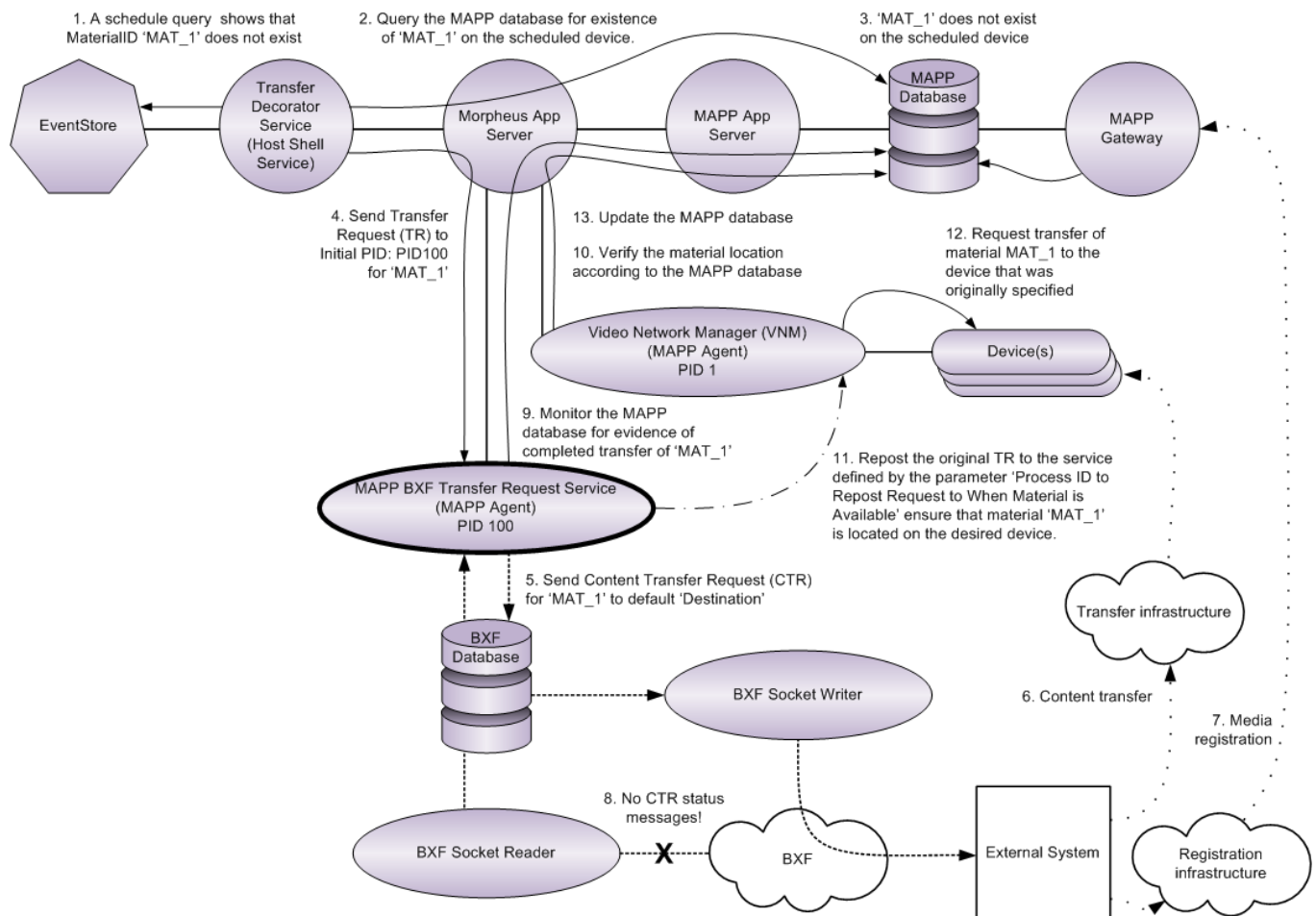
In the case where this service instance (identified in 'Process ID' above) is either unable to process the request or there is another MAPP agent that is more suitable for the task, enter the PID of the alternative MAPP agent to which to post the request. A Transfer Request will only be forwarded for processing to 'Next Process ID' under the following circumstances:

- A fail message has been received from the external system in response to this service sending it a BXF Content Transfer Request. This may be due, for example, to the external system not having access to the material requested - this alternative PID may be more likely to have the Content Transfer processed successfully by a different (PID chain)
- The validity of a transfer 'completed' message received by this service is to be verified
- For performance reasons. Where multiple MAPP Transfer Requests are received for the same material, a single BXF Content Transfer Request is sent to the external system for the MAPP Transfer Request that has the earliest 'required by' entry. Once the material is received, this service will post the original MAPP Transfer Requests to the 'Next Process ID' for duplication and distribution of the material according to the original requirement.



### - Process ID to Repost Request to When Material is Available

If an external system, to which a BXF Content Transfer Request has been sent, is not capable of replying with a request 'completed' or 'failed' message, then this service can only conclude that the request has not been processed. This feature is a component part of an alternative solution to either establish that the content has been transferred or fail the request after the 'Required Time' (as specified in the MAPP database) has expired. The mechanism of this solution is outlined in Figure 2-2, below.



*Fig. 2-2: MAPP BXF Non-Compliance Transfer Messaging Strategy - Logical Topology*

Configure this property with the PID of an alternative service to which this service will repost the original MAPP Transfer Request once it has been established that the content has been transferred to a device other than that which was originally specified, as indicated by an entry having been posted onto the MAPP database. Generally, the configured PID will be that of VNM.

To enable the failing of a request after the 'Required Time' has expired, refer to 'Fail the Request if the Required Time Expires' on the 'Transfer Settings' tab

#### **- Time Between the Process be Executed**

Default: 60 seconds.

The frequency at which this instance of the service will check the MAPP database for the following:

- New messages addressed to its own PID
- The presence of material transfer entries that may require that the original Transfer Request is reposted by this service if the material was transferred to a device other than the one specified (relates to 'Process ID to Repost Request to When Material is Available' property above).

---

Note: The default setting should only be changed on the advice of software engineering.

---

#### **- Time Required Check Interval**

Default: 60 seconds.

A specific timer to set the frequency at which this instance of the service will check the MAPP database for changes to any previous Transfer Requests that are addressed to its own PID.

---

Note: The default setting should only be changed on the advice of software engineering.

---

#### **- Process ID for New Request (0 disable)**

Default: 0 (disabled).

The PID of an alternative destination to that configured for the 'Next Process ID' property, above.

It allows the Transfer Request to be forwarded to a different PID chain that is specifically designed to process particular Transfer Requests.

---

Note: This service will only generate a MAPP Transfer Request and forward it to 'Process ID for New Request' if the related BXF Content Transfer Request was received from an external system.

---

#### **- Next Process Id for Delete / Kill Requests**

Specify a PID that is dedicated to 'Delete' and 'Kill' requests, if one exists. This PID may also be the first in a PID chain.

---

Note: This feature is disabled unless 'Process ID for New Request' is enabled.

---

## Transfer Settings Tab

The screenshot shows the 'MAPP BXF Transfer Request Service Configuration' window with the 'Transfer settings' tab selected. The window has a title bar with standard Windows controls. Below the title bar is a tabbed interface with five tabs: 'General Settings', 'Transfer settings' (active), 'Owner and target device rules', 'Extra information at BXF private section', and 'Automatic'. The 'Transfer settings' tab contains several configuration options:

- ☐ Clean the BXF database for requests that no longer exist
- ☐ Fail the request if the required time expires. To the right is a spin box for 'Time to wait for fail expired request' set to 0, followed by the text 'seconds'.
- Transfer complete message:
  - ☐ Accept transfer complete message
  - ☐ Repost the transfer when the material is not on the final target
- Process ID to repost request to when material is not on final target: A spin box set to 0.
- ☐ Accept transfer with TransferId = 0
- Process ID to repost request to when device is remote: A spin box set to 0.
- Time wait before repost when device is remote: A spin box set to 10, followed by the text 'seconds'.
- ☐ Leave processed date null if an error occur
- Material Type Source special behaviour: A group box containing four radio buttons:
  - ☒ None
  - ☐ Item table
  - ☐ Transfer Requests Parameters field (Schedule Information)
  - ☐ Always \*\*\* BxfUnknown \*\*\*
- ☐ Translate target device and owner
- Device path: A dropdown menu showing 'StorageLocation'.
- ☐ Request only master MaterialId
- Parse expression: A text box containing the regular expression `^(?<base>.+)\/(?<segment>[0-9]+)$`.

At the bottom right of the window are 'Cancel' and 'Save' buttons.

Fig. 2-3: MAPP BXF Transfer Request Service - Transfer Settings Tab

### - Clean the BXF Database for Requests That No Longer Exist

Default: disabled. It is recommended that this feature is enabled.

If this option is ticked, the service will delete entries in the BXF database that relate to MAPP Transfer Request records that no longer exist.

### - Fail the Request if the Required Time Expires

Default: disabled. It is recommended that this feature is enabled.

When enabled, this service will send a 'fail' request to the MAPP database if a Content Transfer has not completed by the 'Required Time'.

Add an extension to the 'Required Time' using the 'Time to Wait for Fail Expired Request' feature below.

---

Note: The 'Required Time' is registered in the MAPP database against the material.

---

**- Time to Wait for Fail Expired Request**

Default: 0 (the recommended value).

An extension to the 'Required Time'.

Transfer Complete Message Pane

The parameters that can be configured in this pane relate only to transfer 'complete' messages.

**- Accept Transfer Complete Message**

Default: disabled.

---

Note: This feature is deprecated and must not be enabled.

---

**- Repost the Transfer when the Material is not on the Final Target**

Default: disabled.

When enabled, this service will post a Transfer Request to the PID specified for 'Process ID to Repost Request to when Material is not on the Final Target' (below), if an associated BXF Content Transfer Request 'completed' message is received indicating that the requested material is on a different device to the one specified in the original Transfer Request that this service retrieved from the MAPP database.

**- Process ID to Repost Request to when Material is not on the Final Target**

Enter the PID for the agent that will process the Transfer Request that has been posted by this service as a result of 'Repost the Transfer when the Material is not on the Final Target' being enabled (above).

**- Accept Transfer with TransferId = 0**

Default: disabled.

When enabled, a BXF Content Transfer 'complete' message will be accepted when received from a BXF service despite it not supporting the 'TransferId' field. The value in the 'MaterialId' field of the message will then be used in order to match it to the original BXF Content Transfer Request message.

---

Note: The 'TransferId' uniquely identifies BXF messages that are associated with a single transaction.

---

---

Note: This feature was created for a very specific customer requirement.

---

## - ProcessID to Repost to When Device is Remote

Note: This parameter is only applicable to 'X & Y' MAPP systems, and is used in conjunction with 'Time Wait Before Repost When Device is Remote' (below).

Enter the process ID of the local Asset Mail agent in order for the synchronisation procedure, described below, to be accomplished. Configure this parameter on both the X system and the Y system.

Given the situation of a Content Transfer Request being sent from both the X and the Y partner systems to an external device, both will receive a transfer complete message, but the content will only ever have been transferred to either system X or system Y. In order to maintain synchronisation, the system that failed to receive the content reissues the Transfer Request to its local Asset Mail agent in order to facilitate the content transfer from the storage device of its remote partner system.

The scenario is detailed further in Figure 2-4.

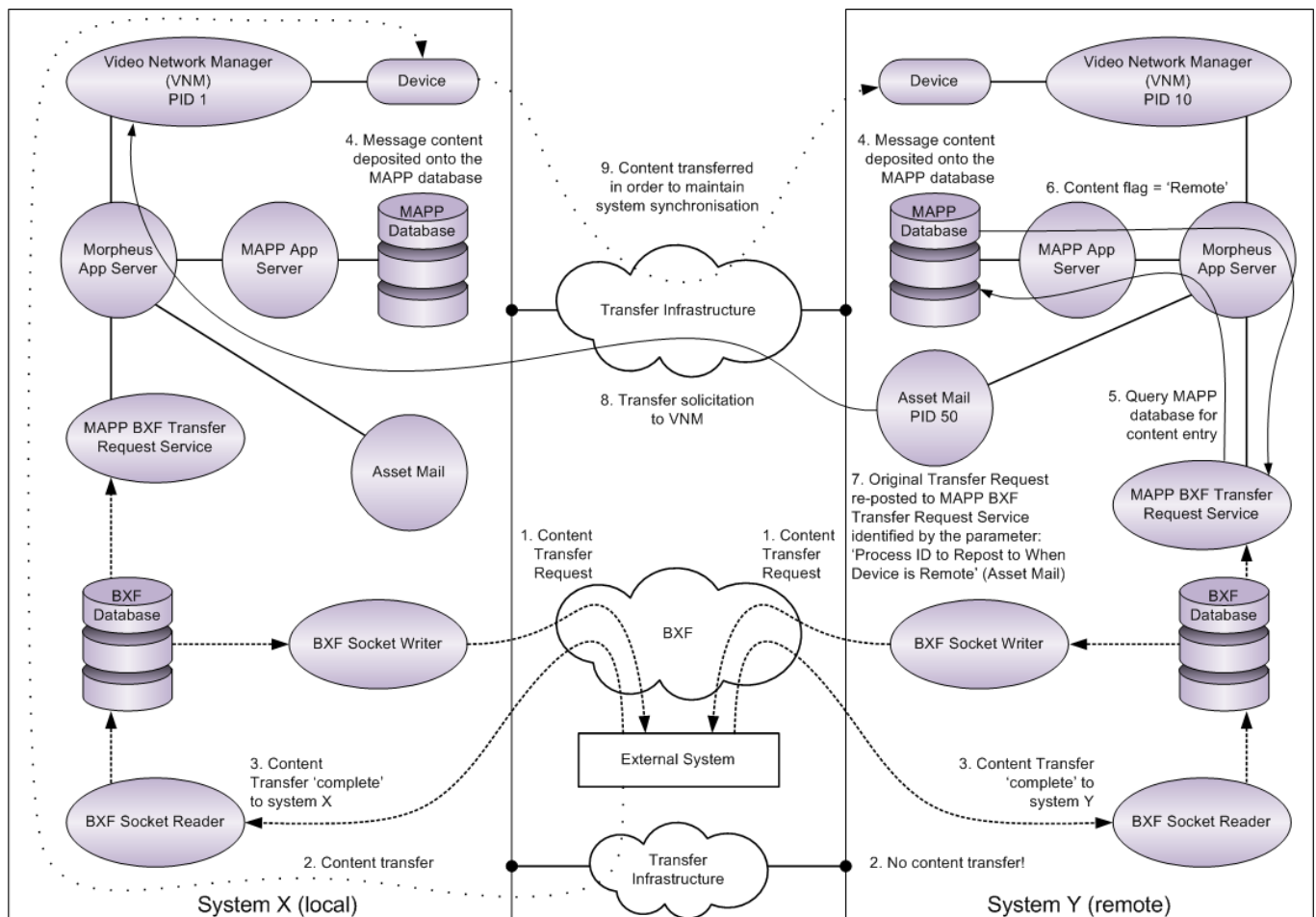


Fig. 2-4: BXF Transfer Requests in X & Y Topologies

#### **- Time Wait Before Repost When Device is Remote**

---

Note: This parameter is only applicable to 'X & Y' MAPP systems, and is used in conjunction with 'ProcessID to Repost to When Device is Remote' (above).

---

Default: 10 seconds (recommended).

For the system, in an X & Y configuration, that receives the transfer 'complete' message but not the content, specify the amount of time that it must wait before sending a Transfer Request to the remote partner system via Asset Mail. It allows time for the content transfer to be registered as 'complete' on the remote system ahead of receiving the new Transfer Request message.

Configure this parameter on both the X system and the Y system.

#### **- Leave Processed Date Null if an Error Occurs**

With this feature disabled, the normal system behaviour is such that the 'Processed Date' field is updated whether or not the Content Transfer is successful or not.

When enabled, the 'Processed Date' field in the BXF Database will be updated for a Content Transfer Request only if it is successful. Should an error occur during the content transfer process then the field is left blank.

---

Note: The 'Accept Transfer Complete Message' parameter must be enabled in order for this function to be operational.

---

---

Note: This feature was created for a very specific customer requirement.

---

### Material Type Source Special Behaviour Pane

The Transfer Decorator Service is responsible for ensuring that content specified in a schedule on the EventStore is available for playout on a particular device, otherwise it will request a material transfer. If the agent that receives the transfer request is the MAPP BXF Transfer Request Service (this service), then it requires specific information in order to populate elements in the BXF message for transmission to an external system. One such element is the `<SpotType>` where the Material Type of non-program content is specified:

```
<NonProgramContent>
  <Details>
    <SpotType>
  </Details>
</NonProgramContent>
```

---

Note: 'Non-Program' content is defined as one of various types that is inserted between the segments of 'Program' content.

---

The `<SpotType>` is mandatory, according to the BXF standard - it defines one of the following types of material supported by this service:

- Commercial
- Junction
- Live Record
- Still

It is possible that no information was supplied from the Transfer Decorator Service regarding the material type, perhaps because none was available from the schedule in the EventStore. This service will therefore attempt to determine the Material Type itself using one of the following selected procedures:

#### **- None**

The recommended setting.

If enabled, the service will cycle once through all of the other options listed below, and in the following order:

1. Item Table
2. Transfer Requests Parameter Field
3. Always **\*\*\*BXFUnknown\*\*\***

#### **- Item Table**

This service will send a message to the MAPP Application Server to request the Material Type from the Item table in the MAPP database.

#### - **Transfer Requests Parameters field (Schedule Information)**

Extract the Material Type from the 'Parameters' field in the Transfer Request Message received from the MAPP BXF Transfer Request Service.

#### - **Always \*\*\*BXFUnknown\*\*\***

The string \*\*\*BXFUnknown\*\*\* is entered for the <SpotType> element in order to indicate to the receiving process that the field is undefined, and can therefore process it accordingly.

#### Main Pane

#### - **Translate Target Device and Owner**

Enables the 'Owner and Target Device Rules' tab in order for any rules configured there to be applied.

#### - **Device Path**

From the drop-down menu, select the format of the path that will be used in a generated message in order to define the device on which the changed content is stored. The format indicates the MAPP database 'Server Type' field from which the values will be extracted in order to define the path. Select from the following options:

- **StorageLocation:** concatenate the value in the DEVICE\_NO field with that of the STORAGE\_LOCATION field in order to create the following path format:

\\DEVICE\_NO/STORAGE\_LOCATION

- **StorageLocationUNC:** use the value in the STORAGE\_LOCATION\_UNC field only, in order to create the following path format:

\\STORAGE\_LOCATION\_UNC

The device path is determined as follows:

1. The received Transfer Request message specifies a Device ID (DEVICE\_NO field in the MAPP database).
2. The selected format specifies the fields in the MAPP database for that Device Id from which to extract the records to create the path.
3. This service queries the Device Id in the MAPP database for the specified fields and inserts the records into the BXF Content Transfer Request message according to the structure of the selected format.



## - Request Only Master MaterialId

Default: disabled.

When enabled, any Content Transfer Request message transmitted by this service will specify that the content with the Master Material ID is required from the destination, even if, for example, the received Transfer Request message specified the Material ID for a soft segment.

The service will automatically determine the Master Material ID from any material that is available on the destination device, and that can be identified from the original Material ID supplied to this service in the Transfer Request message from the Transfer Decorator Service. The regular expression configured for 'Parse Expression' (below) is used to this effect.

## - Parse Expression

---

Note: This feature is disabled unless 'Request Only Master MaterialId' is enabled.

---

Default: `^(?<base>.+?)/(?<segment>[0-9]+)$`

A regular expression that is used with the 'Request Only Master MaterialId' function (above) in order to determine if the material requested in the received Transfer Request is a soft segment. Using the default regular expression, the Master MaterialId is identified as follows:

If the soft segment MaterialID = *ColumboSeries2Episode1/001* then the Master MaterialId is determined to be *ColumboSeries2Episode1*

---

Note: The default regular expression should be modified according to the syntax used in the labelling of soft segments.

---

### Example

If a hyphen is used instead of a forward slash (i.e. *ColumboSeries2Episode1-001*), then the regular expression should be entered as follows:

---

`^(?<base>.+?)-(?<segment>[0-9]+)$`

---

## Destination By Up To Time Required Tab

There is no BXF specification for the prioritisation of Content Transfer Requests by *required time*, therefore any messages transmitted by an external system will be placed in a queue and processed according to the order in which they were received.

This functionality provides the means of associating a number of individual destinations with unique processing targets, overriding the singular default 'Destination' (General Settings Tab) - it was originally created in order to overcome a limitation of in Momentum.

When a BXF Content Transfer Request message is received by Momentum, it will be rejected if it is unaware of the existence of the content, in which case the message is re-transmitted by this service until such a time that it can be processed. This functionality takes advantage of Momentum's behaviour by readdressing the message to a different destination according to the time at which completion of the processing of the message is required - the destinations are all individual Momentum interfaces, treated as different queues (note that Momentum is unaware that the interfaces are treated as different queues).

---

Note: This feature was created for a specific customer requirement. Before implementing a solution, contact software engineering for guidance.

---

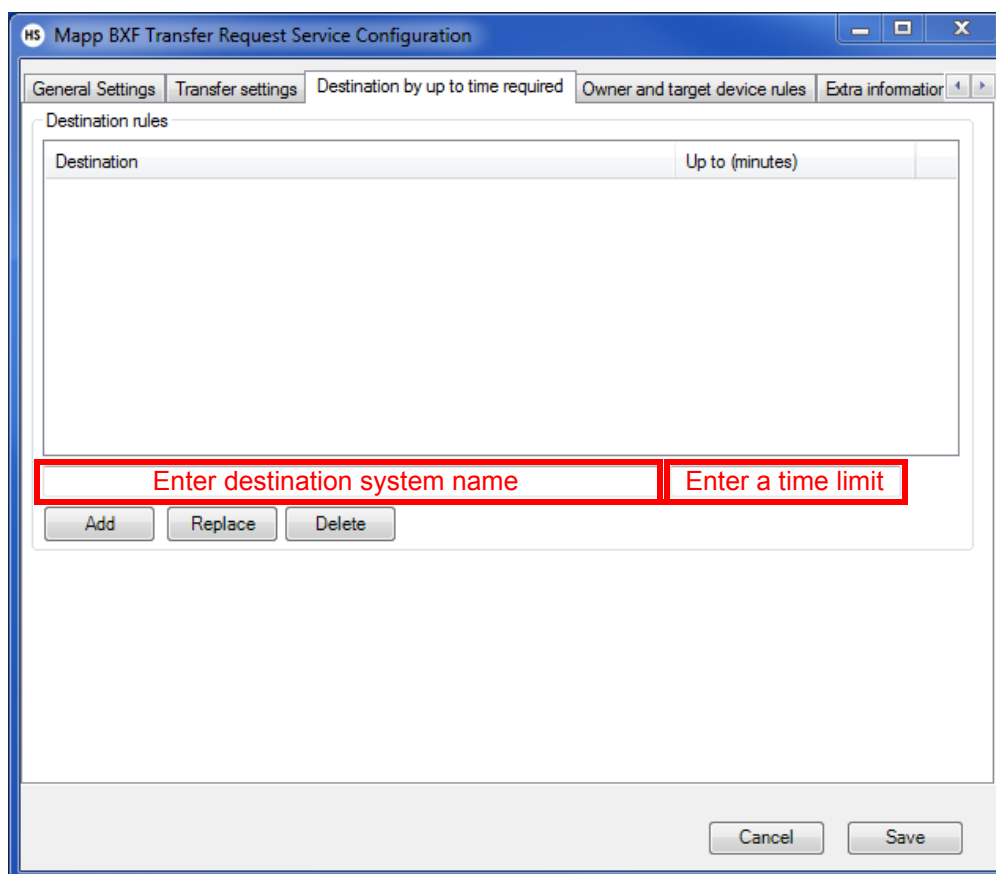


Fig. 2-5: MAPP BXF Transfer Request Service - Destination By Up To Time Required Tab

The queues are configured as 'Destination Rules' using the **Add**, **Replace**, and **Delete** buttons as follows:

- **Add**

- In the 'Destination' field, enter a destination external system name that will become one of the queues (a BXF Socket Writer must be configured for the same 'Destination' associated to an IP address / port number of a Momentum interface).

---

Note: Parameters are case sensitive.

---

- In the 'Up to (minutes)' field, enter a time limit for the queue against which the 'Required Time' can be associated
- Click on **Add**. The rule will appear in the 'Destination Rules' pane.

- **Replace**

To modify an existing destination, click on any part of it and the text boxes will be populated with its current properties. Enter alternative properties then click on the **Replace** button.

- **Delete**

To remove a destination, click on any part of it and click on the **Delete** button.

### Example

Consider the following 'Destination Rules':

Table: 2-6

Destination (Momentum Interface)	Up To (minutes)
System1	30
System2	45
System3	60
System4	90
System5	120

Material in a received Transfer Request message with a 'Required Time' that is more than 120 minutes in the future will use the default 'Destination' (from the 'General Settings' tab) configured in the BXF Content Transfer Request message.

Material in a received Transfer Request message with a 'Required Time' that is more than 90 minutes but less than or equal to 120 minutes in the future will be configured with a destination of 'System5' in the BXF Content Transfer Request message.

Material in a received Transfer Request message with a 'Required Time' that is more than 60 minutes but less than or equal to 90 minutes in the future will be configured with a destination of 'System4' in the BXF Content Transfer Request message. And so on.

Each time Momentum rejects a message, and a 'Required Time' boundary has been crossed, this service will regenerate the message destined for the next higher queue.

## Owner and Target Device Rules Tab

Define rules in order to set alternative target devices in Content Transfer Requests to those specified in received Transfer Request messages. The target device specified is that to which the content should be transferred.

Note: The target device relates to distinct hardware, as listed as an entry in the MAPP database.

Note: The functions on this tab are operational only if 'Translate Target Device and Owner' is enabled (ticked) on the 'Transfer Settings' tab.

Any number of rules can be configured, they are considered in the order in which they appear in the table (from top to bottom), and only the first match is applied. In the absence of a match condition, the default target device is applied to the Content Transfer Request, i.e. that specified by **Device Id (if no rules applied)**.

The screenshot shows a window titled "Mapp BXF Transfer Request Service Configuration" with several tabs: "General Settings", "Transfer settings", "Destination by up to time required", "Owner and target device rules" (which is selected), and "Extra information".

Under the "Owner and target device rules" tab, there is a text field labeled "Device Id (if no rules applied)". Below this is a label "Define rules for owner and target device based on parameters at TRANSFER\_REQUEST.PARAMETERS".

A table titled "Rules" is present, with the following columns: "Parameter name", "Parameter value", "Owner", and "Target device". The table is currently empty.

Below the table, there are three buttons: "Add", "Replace", and "Delete".

At the bottom right of the window are "Cancel" and "Save" buttons.

A red rectangular box highlights the column headers of the "Rules" table: "Parameter Name", "Parameter Value", "Parameter Owner", and "Target Device".

Fig. 2-6: MAPP BXF Transfer Request Service - Owner and Target Device Rules Tab

### - Device Id (if no rules applied)

A default target device that is applied in the event that there is no match to any rule. Specify the default target device to be entered into the Content Transfer Request message, replacing that which was received in the Transfer Request message.

## - Rules

### • Add

Configure all of the properties, as follows:

#### a) *Parameter Name*

---

Note: Parameters are case sensitive.

---

Enter the parameter name against which to match in the received Transfer Request message. Parameters against which to match are those defined in the 'Parameters' field in the 'Transfer Request' table on the MAPP database.

---

Note: Only the value attributed to <Parameter Name> is validated by this service when the **Add** or **Replace** buttons are pressed.

---

#### b) *Parameter Value*

---

Note: Parameters are case sensitive.

---

Specify the value of 'Parameter Name' (above) against which to match. Parameters against which to match are those defined in the 'Parameters' field in the 'Transfer Request' table on the MAPP database

#### c) *Owner*

Set a value (string) that will be entered into the 'PrivateInformation' element of the BXF Content Transfer Request message upon a parameter / value pair match condition.

---

Note: The value of <Owner> is only of relevance to the external system to which the Content Transfer Request message will be sent, allowing the message to be processed accordingly.

---

#### d) *Target Device*

Specify the destination device, as listed in the MAPP database, to which the content should be transmitted; the target device specified will be entered into the Content Transfer Request, replacing that which was stipulated in the received Transfer Request.

#### e) Click on **Add**. The rule is displayed in the 'Rules' pane.

### • Replace

To modify an existing rule, click on any part of it and the text boxes will be populated with its current properties. Enter alternative properties then click on the **Replace** button.

- **Delete**

To remove a rule, click on any part of it and click on the **Delete** button.

A rule operates in the following manner:

IF *<Parameter Name>* AND *<Parameter Value>* exist in the Transfer Request  
THEN *<Owner>* is entered into the 'PrivateInformation' element of the Content Transfer Request message  
AND the *<Target Device>* is entered into the Content Transfer Request, replacing the original that was stipulated in the received Transfer Request.

---

Note: All fields must be populated for the rule to be operational.

---

## Extra Information at BXF Private Section Tab

Define the 'Parameters' fields from the 'Transfer Request' table on the MAPP database, the values of which must be inserted into the BXF 'PrivateInformation' element of the Content Transfer Request message.

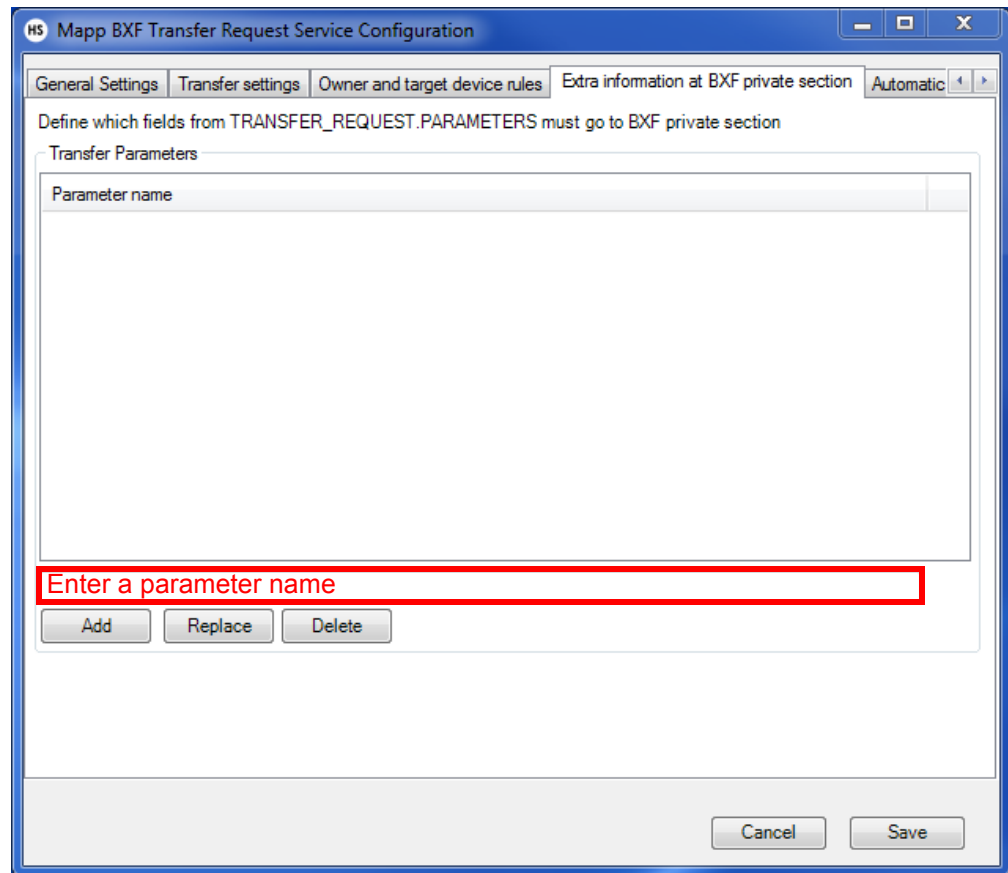


Fig. 2-7: MAPP BXF Transfer Request Service - Extra Information at BXF Private Section Tab

### - Add / Replace / Delete

- **Add**

- a) Enter a field from TRANSFER\_REQUEST.PARAMETERS IN THE MAPP database.

---

Note: Parameters are case sensitive.

---

- b) Click on **Add**. The parameter is entered into the 'Parameter Name' pane.

- **Replace**

To modify an existing transfer parameter, click to select it and it will appear in the text box. Modify the property name then click on the **Replace** button.

- **Delete**

To remove a parameter, click to select it and click on the **Delete** button.

## Automatic Detect Completed Requests Tab

Functionality to allow this service, whilst processing a Transfer Request message, to investigate the existence of a record of the content existing on one or more local devices at the following stages of the content transfer process:

- Before transmitting a Content Transfer Request message to an external system  
Should this occur, then the Transfer Request message is re-posted to the agent specified in 'Process ID to Repost Request to When Material is Available' (refer to General Settings Tab)
- Whilst the external system is processing a Content Transfer Request message transmitted by this service for the same content

Note: The existence of a record of the content can be investigated based upon either a fixed device type or a list of specific device Id's.

### - Automatic Detect Completed Requests

Tick this checkbox in order to enable the functionality.

Once enabled, all received Transfer Requests will be held by this service until it is discovered that the content is located on the target device defined by 'Device Type' or 'Device ID'.

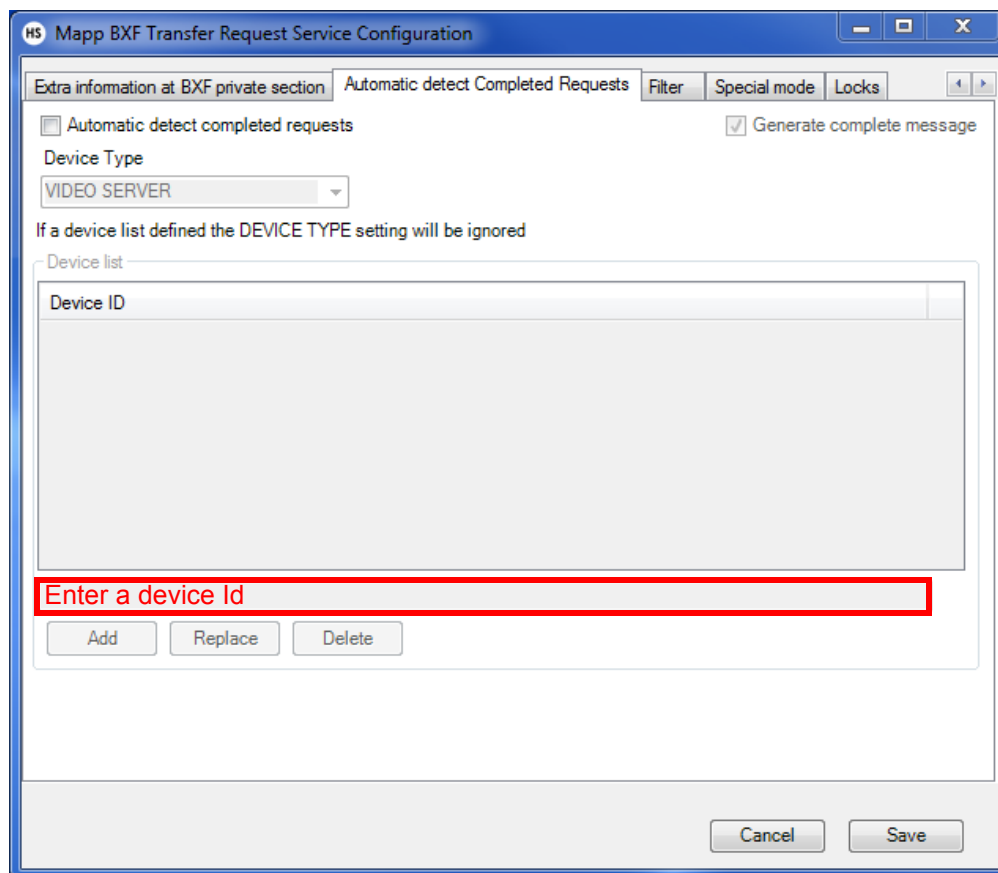


Fig. 2-8: MAPP BXF Transfer Request Service - Automatic Detect Completed Requests Tab



### - Generate Complete Message

When ticked, a BXF Content Transfer 'complete' message will be transmitted to an external system under the following conditions:

- The external system has completed the transfer request subsequent to this service failing to discover the existence of a record of the content existing on a local device
- This service has been successful in discovering the existence of a record of the content locally. The 'complete' message is sent regardless of the state of the content transfer process, i.e. before or after sending the Content Transfer Request, and even if the external system has begun processing it.

---

Note: This is a proprietary mechanism and does not therefore adhere to the BXF messaging protocol.

---

### - Device Type

From the drop-down menu, select one of the following device types on which to base the search for a record of the content existing locally:

- VIDEO SERVER (default)
- GRAPHICS ARCHIVE

The supported device types are two of those that are defined in the 'Type' field of the MAPP database.

---

Note: If the 'Device ID' table (below) is populated, then the selected 'Device Type' is ignored.

---

### - Device ID

The IDs of devices that this functionality will interrogate in order to investigate the existence of a record of the content existing locally.

### - **Add / Replace / Delete**

Add, replace, or delete the IDs of devices that this functionality will interrogate in order to investigate the existence of a record of the content existing locally.

- **Add**

- a) Enter a device ID.

---

Note: Parameters are case sensitive.

---

- b) Click on **Add**. The device ID is displayed in the 'Device ID' pane.

- **Replace**

To modify an existing device ID, click to select it and it will appear in the text box. Modify the ID then click on the **Replace** button.

- **Delete**

To remove a device ID, click to select it and click on the **Delete** button.

## Filter Tab

This feature allows the service to filter out Transfer Requests for material types.

Define the Material Types from a received Transfer Request for which this service will generate a Content Transfer Request - upon a no-match condition the Transfer Request will be failed.

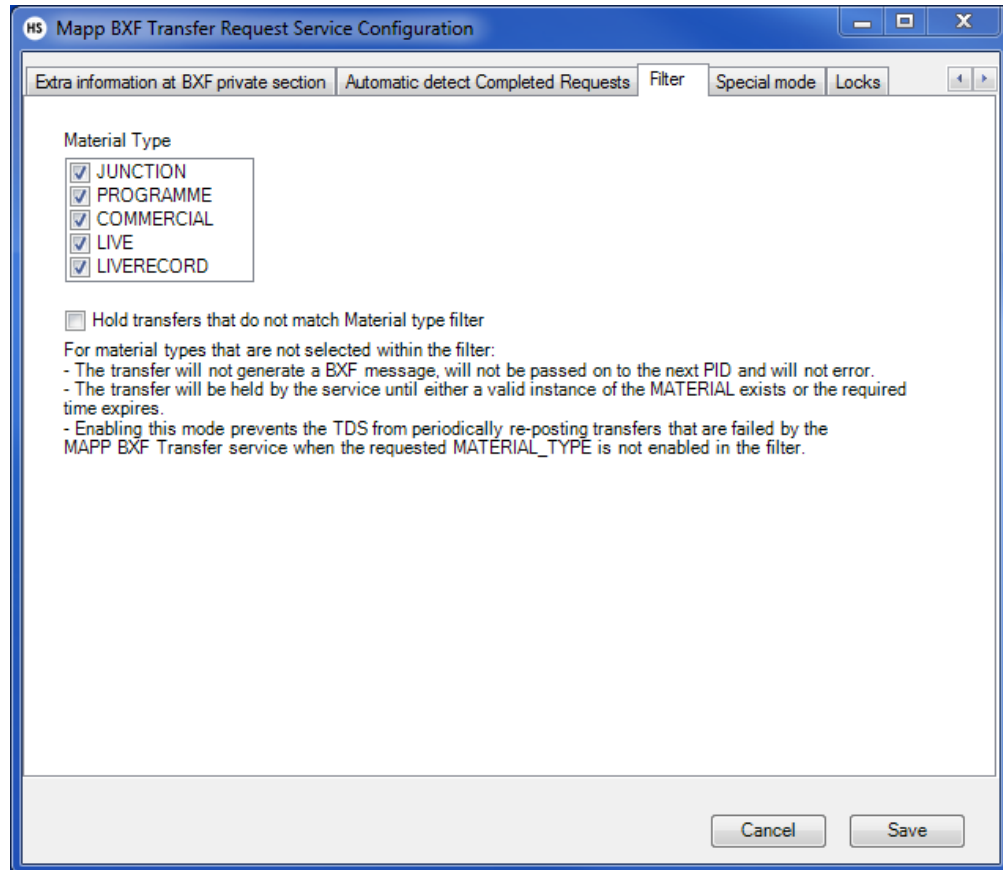


Fig. 2-9: MAPP BXF Transfer Request Service - Filter Tab

### - Material Type

Tick each Material Type against which to perform a match. Select from the following:

- JUNCTION
- PROGRAMME
- COMMERCIAL
- LIVE
- LIVERECORD

**- Hold Transfers that do not Match Material Type Filter**

Enable this feature in order to prevent the Transfer Decorator Service from re-posting transfers that have been failed by this service when the requested Material Type is not selected on the filter list.

When enabled (ticked), for any no-match condition on a Material Type, this service will only process a received Transfer Request message if a record of an instance of the requested material is posted on the MAPP database by another service.

This service will monitor the MAPP database at the interval specified in 'Time Between the Process be Executed' for evidence of a record of the requested material being posted on the MAPP database [General Settings Tab].

If the 'Required Time' in the Transfer Request message expires then this service will fail the message.

## Special Mode Tab

### Import Metadata Pane

A function to allow only the metadata to be ingested from a received BXF Content Transfer Request, ignoring the request to transfer the material.

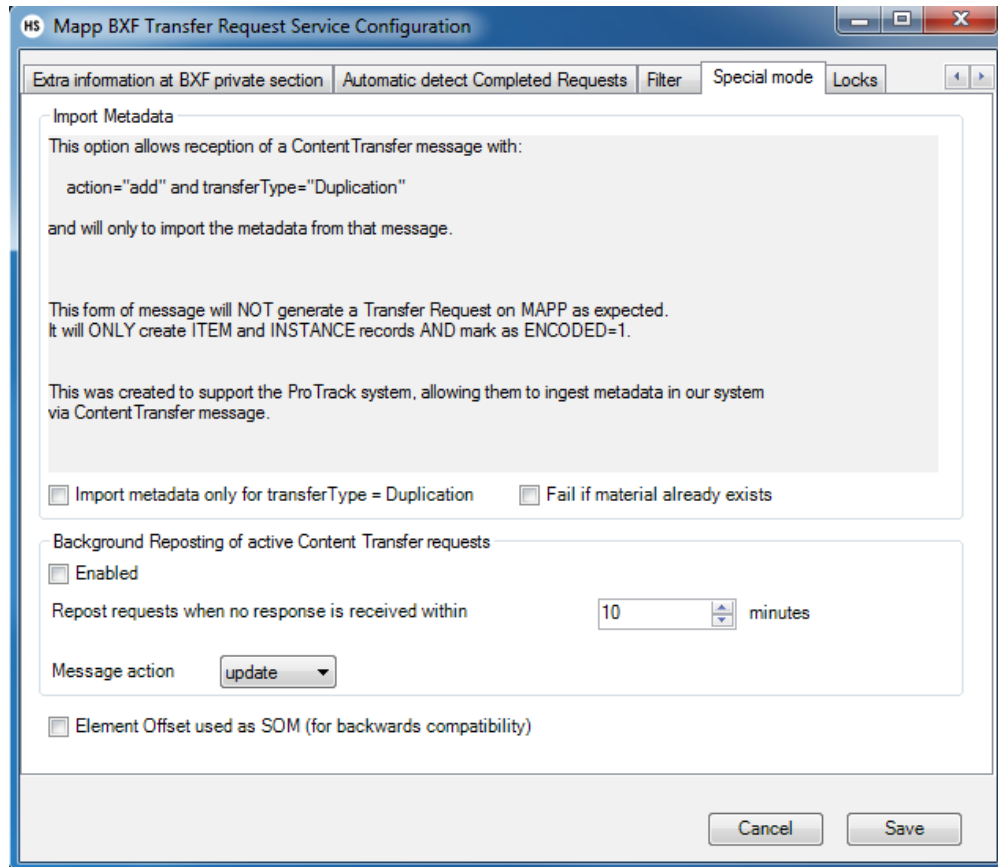


Fig. 2-10: MAPP BXF Transfer Request Service - Special Mode Tab

When enabled, if a Content Transfer Request, received from an external system, contains both of the attributes listed immediately below, then this service will not generate a Transfer Request, but it will solicit the MAPP database to create a record for the requested material.

- *action* = add
- *transferType* = duplicate

---

Note: This is a proprietary mechanism and does not therefore adhere to the BXF messaging protocol.

---

**- Import Metadata only for TransferType=Duplication.**

Enable the 'Import Metadata' functionality.

**- Fail if Material Exists**

Send a Content Transfer Request 'fail' message to the external system if a record of the requested material already exists in the MAPP database.

**Background Reposting of Active Content Transfer Requests Pane**

When enabled, unless a 'complete' or a 'fail' message is returned from the external device to which this service has transmitted a Content Transfer Request, then the same Content Transfer Request will be re-posted upon the expiry of the value attributed to 'Repost Requests When No Response is Received Within'.

---

Note: The following properties must be configured when this feature is enabled:

- 'Repost Requests When no Response is Received Within'
  - 'Message Action'
- 

**- Enable**

Enable this feature.

**- Repost Requests When no Response is Received Within**

Default: 10 minutes (recommended).

An amount of time after which this service will re-post a Content Transfer Request according to the conditions for 'Background Reposting of Active Content Transfer Requests', above.

---

Note: The feature 'Background Reposting of Active Content Transfer Requests' must be enabled in order for this property to be effective.

---

**- Message Action**

The attribute that will be entered into the Content Transfer Request when it is re-posted according to the conditions for 'Background Reposting of Active Content Transfer Requests', above.

Select one of the following available options:

- Update (default)
- Add

---

Note: The feature 'Background Reposting of Active Content Transfer Requests' must be enabled in order for this property to be effective.

---

**- Element Offset Used as SOM (for backwards compatibility)**

This feature was added in Morpheus v5.9.2

Default: disabled (un-ticked)

For soft segments, the original iteration of the Morpheus software to support the content `<offset>` element in a BXF message differed from the specification of the standard, in that it was interpreted as the Start Of Material (SOM). The `<offset>` element in the BXF message would therefore have to contain a SOM for each soft segment, and not the offset from the main clip as would be expected.

The software was modified in v5.2.3 in order to reflect the correct interpretation of the BXF standard, i.e. the `<offset>` element now defines the start point of each soft segment relative to the beginning of the main clip.

This feature provides the ability to switch between both modes. When enabled, the original interpretation of the `<offset>` element is supported in order to allow Morpheus systems to be migrated from pre-v5.2.3 software to v5.9.2 onwards without the need for customers to also have to modify their BXF traffic systems concurrently. Once the BXF traffic systems have been amended in order to support the correct interpretation of the `<offset>` element, then this function can be disabled.

## Locks Tab

This lock mechanism is an extension to the MAPP locks facility, providing the means of securing local material from being either moved or deleted by an automation system when it is required to be accessed by an external system. The lock is applied by this service on behalf of an external system. A possible use case is when a resource required to transform or convert the material (such as a Momentum system) is only available externally.

This service will inform the local MAPP database to enter lock records for the material specified in a received Transfer Request for one or more specific storage devices. The record(s) will be entered into the MAPP database regardless of whether or not the material actually exists on the specified storage devices. A lock is removed once this service has received a Content Transfer Request 'complete' message from the external system upon completion of its task.

A lock does not prevent the reading or duplication of the material.

Note: A lock applied to a record of material in the MAPP database does not prevent the actual material from being deleted on the storage device (e.g. manually).

The mode of operation of this feature is detailed in Figure 2-11.

The dashed lines indicate the BXF message transmission path, whereas the solid lines represent the Morpheus / MAPP delivery mechanism.

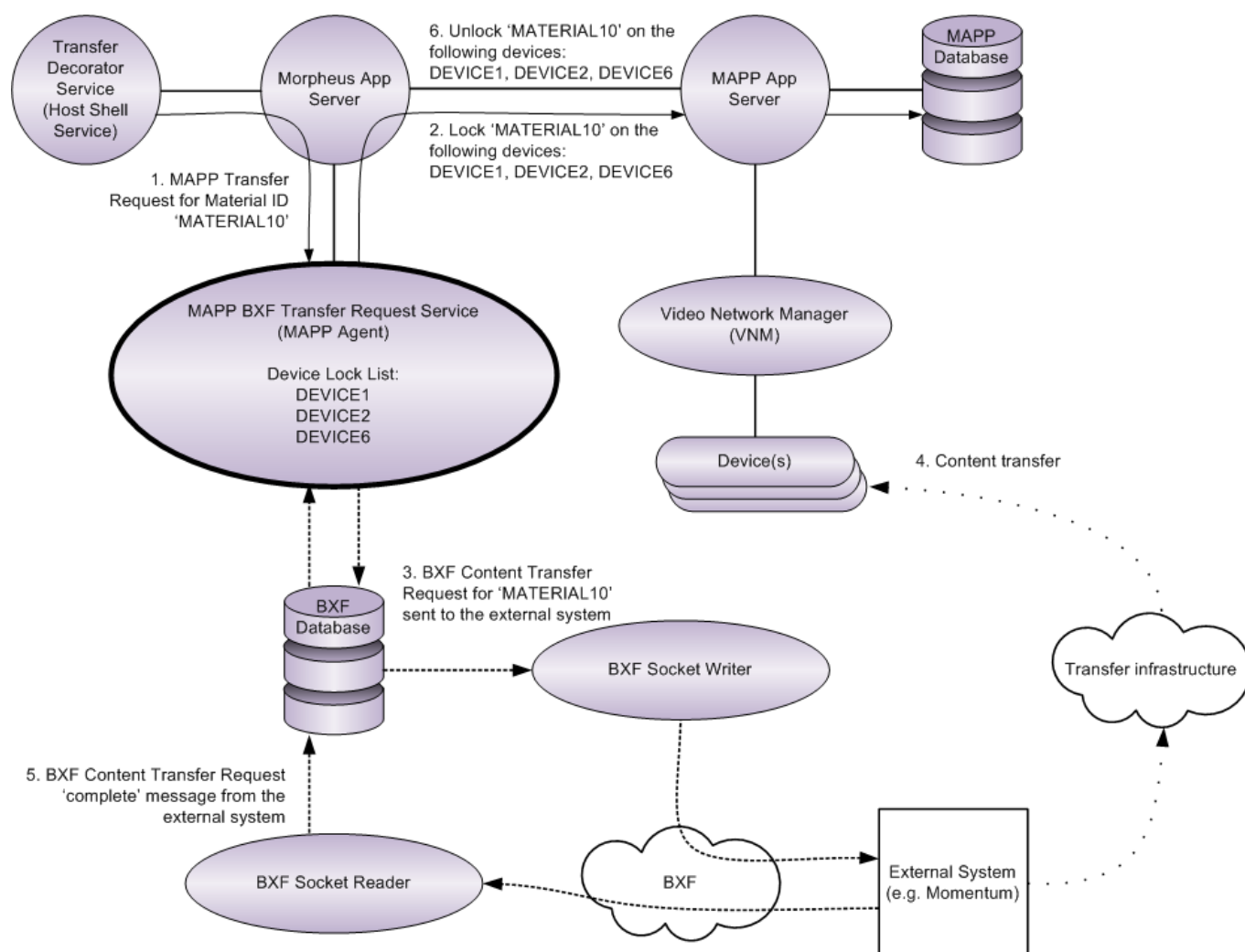


Fig. 2-11: MAPP BXF Transfer Request Service - Locks Operation



### - Lock Transfer Source

Default: disabled.

Tick the checkbox in order to enable the locks functionality.

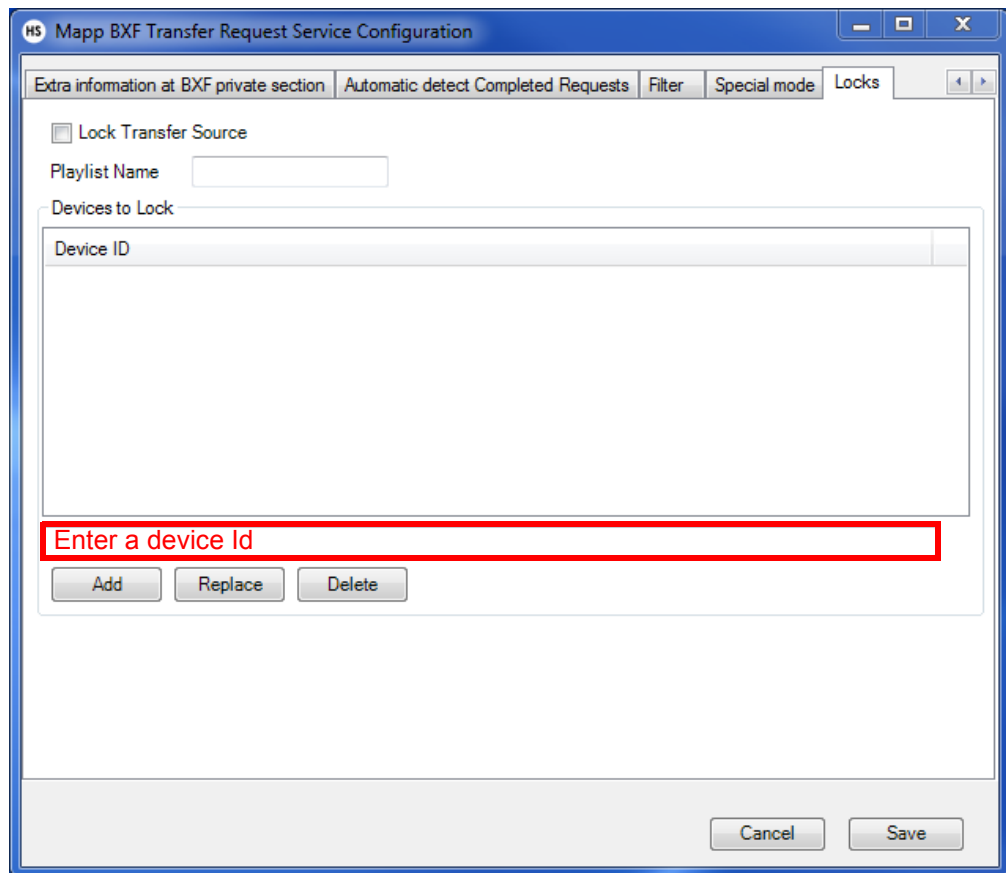


Fig. 2-12: MAPP BXF Transfer Request Service - Locks Tab

### - Playlist Name

This is a required field but is has no active purpose. Enter any text string.

### - Devices to Lock

The devices for which a lock record must be created in the MAPP database for the material specified in the received Transfer Request, regardless of whether or not the material actually exists on those devices.

### - **Add / Replace / Delete**

Add, replace, or delete Device ID's for which this functionality will inform the local MAPP database to enter lock records.

- **Add**

- a) Enter a device ID.

---

Note: Parameters are case sensitive.

---

- b) Click on **Add**. The device ID is displayed in the 'Device ID' pane.

- **Replace**

To modify an existing device ID, click to select it and it will appear in the text box. Modify the ID then click on the **Replace** button.

- **Delete**

To remove a device ID, click to select it and click on the **Delete** button.



## **Grass Valley Technical Support**

For details of our Regional Customer Support Offices please visit the Grass Valley website and navigate to Support> Contact Support.

<https://www.grassvalley.com/support>

## **UK Office**

The department is staffed from 9.00am to 5:30pm Monday to Friday (excluding UK public holidays).

Outside these times, calls will be delivered to voicemail for follow up on the next working day.

Additional support is available outside these hours by purchasing a support contract, details of which are available from the Grass Valley website and through the account manager.

Customers with an existing support contract should call their personalised number, which can be found in the contract, and be ready to provide the contract number and details.