

Selling and Fulfillment Foundation

Properties Guide

Release 9.0

Last updated in HF4

August 2010

Sterling Commerce
An IBM Company

© Copyright 2010 Sterling Commerce, Inc. All rights reserved.
Additional copyright information is located on the documentation library:
<http://www.sterlingcommerce.com/Documentation/MCSF90/CopyrightPage.htm>

Contents

| | |
|--|-----------|
| Configuring Properties | 5 |
| Initial Settings for Properties Files | 6 |
| Overriding Properties | 7 |
| Using the Property Parameters File (sandbox.cfg) | 7 |
| Using the Customer_Overrides.properties File | 8 |
| Locating the Property File Name Prefix | 8 |
| Example 1: Overriding a property from the yfs.properties.in file | 9 |
| Example 2: Overriding a property from the dbclassCache.properties file | 9 |
| Properties for LDAP User Authentication | 11 |
| Logging Configuration | 13 |
| Masking Sensitive Information During Logging Using log4j | 14 |
| Using Custom Layout and Filter | 14 |
| Setting Up Regular Expressions | 15 |
| Enabling Different Properties for Individual Processes | 17 |
| Configuring Cache for Catalog Search Index | 5 |
| Property Files Reference | 7 |
| Making Changes to Properties | 7 |
| Setting the Database Connection Properties | 7 |
| Datasource Connection Pooling Property | 7 |
| yfs.properties* | 8 |
| dbclassCache.properties.* | 46 |
| Sandbox.cfg | 47 |
| Index | 56 |

Configuring Properties

Property files contain properties that control the operation of Selling and Fulfillment Foundation. By modifying the values of these properties, you can customize Selling and Fulfillment Foundation to suit your business and technical needs.

After installing Selling and Fulfillment Foundation, most property and script files do not need any further configuration for basic operation of the system. However, if you want to customize any specific operations—for example, setting a different logging level—you will need to edit (and in some cases, create) certain property or.xml files.

In general, changes to properties are not made in the specific property files themselves; changes are made to the customer_overrides.properties file or sandbox.cfg.

Detailed information about each configurable property in sandbox.cfg, yfs.properties*, management.properties, and dbclassCache.properties is located in "[Property Files Reference](#)".

Notes:

- ◆ Change only the properties included in this appendix. Sterling Commerce does not support changes to any other properties.
- ◆ This chapter contains the information required for property configuration to use with databases, agent servers, LDAP servers, and logging. This property configuration is required during Selling and Fulfillment Foundation installation, as described in the *Selling and Fulfillment Foundation: Installation Guide*.
- ◆ If you are upgrading from a prior release, see the *Selling and Fulfillment Foundation: Upgrade Guide* that applies to your implementation **before** continuing with the setup of Selling and Fulfillment Foundation. Changes have been made to the properties files in the Release 8.2, so you must read the *Selling and Fulfillment Foundation: Upgrade Guide* for information about how this change impacts your system.

Initial Settings for Properties Files

In Selling and Fulfillment Foundation, property files are generated when the `<INSTALL_DIR>/bin/setupfiles.sh` (or `setupfiles.cmd`) script is run from the corresponding initial settings files. The initial settings files are shipped with the product and are present in the same `<INSTALL_DIR>/properties` directory.

The `*.in` files contain variable placeholders (parameters) for properties that contain installation- or environment-specific information (such as database host, port, or credential information).

The installer gathers these parameters (either interactively or using a silent installation parameter file) and places this initial configuration information into a special parameters file called `sandbox.cfg`. Using the parameters contained in the `sandbox.cfg`, the `*.in` files are processed by the `setupfiles` script to create the final properties files that are used by the product during runtime.

Sterling Commerce does not recommend that you modify or change any properties in files ending with `.in` because newer versions or patches of the product will overwrite your changes. Sterling Commerce also does not recommend that you change a property file that has a corresponding `.in` file because the `setupfiles` script will re-create the properties file again, thus causing you to lose your changes. The following section describes overriding properties.

Overriding Properties

Even though it is recommended that you do not change property files directly, situations do occur that require property changes. For these situations, use the `customer_overrides.properties` file, `sandbox.cfg` file, and `log4j.custom.xml` file, which are described below.

- ◆ `customer_overrides.properties` – If you need to change the value of a property that is not parameterized in the `.in` file(s), you can override that property by adding an entry for it to the `customer_overrides.properties` file. Making changes to this override file instead of individual property files ensures that none of your changes are lost in an upgrade or patch installation, and lets you see most of the overrides in your configuration at a glance, rather than having to check each property file individually. This file is not delivered with the product; you must create it the first time you have a need for it.
- ◆ `sandbox.cfg` – Contains name-value parameters that are merged with each `*.in` file to create the final properties files. For values that are parameterized in the `*.in` files, you can supply that parameter in the `sandbox.cfg` file.
- ◆ `log4j.custom.xml` – For changes to logging properties, you create a file called `log4j.custom.xml`. You make changes to basic logging properties in this file. There are additional logging properties in the `yfs.properties.in` file. To make changes to these, you create entries in the `customer_overrides.properties` file.

Using the Property Parameters File (`sandbox.cfg`)

The `<INSTALL_DIR>/properties/sandbox.cfg` file contains name-value parameters that are merged with each `*.in` file to create the final properties files. You can examine any `*.in` file and if you find a value that is parameterized, you can supply that parameter in the `sandbox.cfg` file. A parameter is contained within the ‘&’ and ‘;’ characters. For example, the `jdbc.properties.in` file contains the following property:

```
oraclePool.user=&ORA_USER;
```

The `&ORA_USER;` signifies a parameter. If the `sandbox.cfg` file contains the entry:
`ORA_USER=oracle`, the resulting `jdbc.properties` file will contain the following property:

```
oraclePool.user= oracle
```

You can edit the information in the `sandbox.cfg` file at any time to change values that have been created by the installer or to reflect changed setup parameters.

Most of the parameters in the `sandbox.cfg` file and the `*.in` files are not used at runtime by the product. Consequently, if you change a parameter in the `sandbox.cfg` file, you must run the `setupfiles` script so that the runtime property files are re-created with the updated values.

See "[Property Files Reference](#)", for a list of properties that can be changed and a description of each.

Using the `Customer_Overrides.properties` File

The `sandbox.cfg` parameters let you change the properties defined with variables. However, if you need to change the value of a property that is not parameterized in the `.in` file(s), you can override that property by creating a special file called `customer_overrides.properties`.

For each property that you want to override, you must have the following information:

- ◆ `PROPERTY_FILE_NAME_PREFIX` - Name used in the `servers.properties` file to reference the actual property file.
- ◆ `PROPERTY_NAME` - The name of the property as used in the specified property file.
- ◆ `PROPERTY_VALUE` - The value you want to assign to the property.

These values are used to create an entry in `customer_overrides.properties` that looks similar to the following example:

```
PROPERTY_FILE_NAME_PREFIX.PROPERTY_NAME=PROPERTY_VALUE
```

Locating the Property File Name Prefix

To find the `PROPERTY_FILE_NAME_PREFIX` for a property:

1. If you do not know the name of the file containing the property you are overriding, find the file by searching for the property in the `<INSTALL_DIR>/properties` directory. For example, if you want to find the file that contains the `api.security.token.enabled` property, search the `<INSTALL_DIR>/properties` directory for `api.security.token.enabled`. In this case, you find the `yfs.properties` file.
1. In the `<INSTALL_DIR>/properties` directory, locate the `servers.properties` file and open it in a text editor.
2. In the `servers.properties` file, find the entry for the property file that you located in step 1. In the example for step 1, you found the `yfs.properties` file. In this case, find an entry for the `yfs.properties` file in the `servers.properties` file.

Note: Extensions for property files are not listed in the `servers.properties` file. For example, `yfs.properties_ycs_ext` is listed as `yfs.properties` in the `servers.properties` file.

3. The part of the entry before the equal sign (=) is the prefix that you will use in `customer_overrides.properties`. Make note of it.

For example, locate the entry for `yfs.properties` in `servers.properties`:

```
yfs=<INSTALL_DIR>/properties/yfs.properties
```

`yfs` is the prefix for the `yfs.properties` property file.

To create an entry in the `customer_overrides.properties` file:

1. In the `<INSTALL_DIR>/properties` directory, locate (or create if necessary) the `customer_overrides.properties` file.
2. Open the `customer_overrides.properties` file in a text editor.
3. Add the properties that you want to override, using the following format:
`PROPERTY_FILE_NAME_PREFIX.PROPERTY_NAME=PROPERTY_VALUE`
4. Save and close the `customer_overrides.properties` file.
5. Stop your Selling and Fulfillment Foundation server.
6. Rebuild the EAR, following the instructions for your application server type, as described in the *Selling and Fulfillment Foundation: Installation Guide*.
7. Restart your Selling and Fulfillment Foundation server and all agent and integration servers.

Example 1: Overriding a property from the `yfs.properties.in` file

1. Check `servers.properties` to find the `PROPERTY_FILE_NAME_PREFIX`. In `server.properties`, the line for `yfs.properties.in` is:
`yfs=<INSTALL_DIR>/properties/yfs.properties`
`yfs` is the `PROPERTY_FILE_NAME_PREFIX`.
2. The `yfs.properties` file contains a property called `yfs.smtp.connectionpool.enable`. For an override entry, the `PROPERTY_NAME` value would be `yfs.smtp.connectionpool.enable`. So far, the entry for `customer_overrides.properties` so far would be:
`yfs.yfs.smtp.connectionpool.enable`
3. The default value for `yfs.smtp.connectionpool.enable` is "true". To override this default, you would add the value of "false" to the entry. The completed override entry for the example is shown below.
`yfs.yfs.smtp.connectionpool.enable=false`

Note: For more information about properties in the `yfs.properties` files that can be changed, see "[Property Files Reference](#)".

Example 2: Overriding a property from the `dbclassCache.properties` file

1. Check `servers.properties` to find the `PROPERTY_FILE_NAME_PREFIX`. In `server.properties`, the line for `dbclassCache.properties` says:
`dbclassCache=<INSTALL_DIR>/properties/dbclassCache.properties`
`dbclassCache` is the `PROPERTY_FILE_NAME_PREFIX`.
2. The `dbclassCache.properties` file contains a property called `sci.globalcache.object.size`. For an override entry, the `PROPERTY_NAME` value would be `sci.globalcache.object.size`. So far, the entry for `customer_overrides.properties` so far would be:
`dbclassCache.sci.globalcache.object.size`

3. The default value for `sci.globalcache.object.size` is 10000. To override this default, you would add the new value (in this example, 15000) to the entry. The completed override entry for the example is shown below.

```
dbclassCache.sci.globalcache.object.size=15000
```

Notes:

- ◆ For more information about properties in the `dbclassCache.properties` file that can be changed, see "[Property Files Reference](#)".
- ◆ You do **not** need to run the `setupfiles` script after making a change to the `customer_overrides.properties` file, because no parameters are being replaced. However, you do need to stop and restart your Selling and Fulfillment Foundation server and all agent and integration servers for the changes to take effect.
- ◆ Remember to comment the `dbtype`, `driverclass` and `jdbcurl` that does not pertain to your configuration.

Properties for LDAP User Authentication

This section assumes you understand how LDAP servers work. Sterling Commerce also recommends that you read the following documents on LDAP technology:

- ◆ W. Yeong, T. Howes, and S. Kille, *RFC 1777 - Lightweight Directory Access Protocol*. March 1995. Available at <http://www.faqs.org/rfcs/rfc1777.html>.
- ◆ Mark Wilcox, *Implementing LDAP*. Wrox Press, 1999.

By default, all authentication is performed against the Selling and Fulfillment Foundation database. When a user enters a login ID and password, it is validated against the login ID and password that is stored in the database. This requires the administrator of the Selling and Fulfillment Foundation system to set up login IDs and passwords for each user.

Alternatively, the Application Consoles support LDAP-based user authentication. You may choose to use an LDAP server for authentication. When using LDAP, the users, user groups, and access control must be set up in the Selling and Fulfillment Foundation system.

Selling and Fulfillment Foundation also supports password expiration through LDAP. Your custom code for user authentication is interfaced with the Selling and Fulfillment Foundation authentication mechanism. If your custom code contains `ExpireInDays` with a numeric value of `<x>`, then a message to reset the password appears in the Selling and Fulfillment Foundation home page. If the map contains `ChangePasswordLink` then the message contains a link to the location specified. Clicking on the link opens a new window with the given `ChangePasswordLink`.

Since the various implementations of LDAP, handle password expiration differently a sample `YFSLDAPAuthenticator` is modified to provide an example of one particular implementation. This is located in the `<INSTALL_DIR>/xapidocs/code_examples/java` directory.

To set properties for LDAP-based authentication:

1. Install the LDAP server (see the installation instructions from your LDAP server vendor).
2. If a JAAS-compliant provider is used, create a JAAS configuration file with the following lines:

```
LDAP
{
  // refer to the JAAS compliant service provider for the login
  module details.
  <Class Name of the Login Module as specified by the Security
  provider> required
  debug=true;
};
```

3. In your `customer_overrides.properties` file, specify the LDAP properties described in the following table.

| Property | Description |
|--|---|
| In the customer_overrides.properties file, specify: | |
| yfs.yfs.security.authenticator | Develop a new class that implements the <code>com.yantra.yfs.japi.util.YFSAuthenticator</code> interface and set the new classname as value for this property. Note: The Selling and Fulfillment Foundation provides a sample <code>com.yantra.yfs.util.YFSLdapAuthenticator</code> class that you can use for reference. |
| yfs.yfs.security.ldap.factory | If the default implementation is used, this property specifies the LDAP context factory classname as in your LDAP Server configuration. Set this property value to <code>com.sun.jndi.ldap.LdapCtxFactory</code> . |
| yfs.yfs.security.ldap.url | If the default implementation is used, this property specifies the URL used to access your LDAP Server. For example, <code>yfs.security.ldap.url=ldap://MyServer:800</code> . |
| yfs.yfs.security.ldap.o | If the default implementation is used, this property specifies the Selling and Fulfillment Foundation organization in your LDAP Server configuration. |
| yfs.yfs.security.ldap.ou | If the default implementation is used, this property specifies the Selling and Fulfillment Foundation organizational unit in your LDAP Server configuration. |
| yfs.yfs.jaas.loginmodule | If using JAAS, set this property value to LDAP. |
| yfs.yfs.security.authenticator | If using JASS, set this property value to <code>com.yantra.interop.services.security</code> . |
| WebLogic startWLS startup file | |
| -Djava.security.auth.login.config | If you are using JAAS and WebLogic, specify the full path to your JAAS configuration file. |
| In the Applications Manager | |
| Configure organizations, organization units, and users. | All the users who need to access the Selling and Fulfillment Foundation system must be set up under the LDAP server. All Selling and Fulfillment Foundation users must belong to the same organizational unit. |

Logging Configuration

Selling and Fulfillment Foundation includes basic logging functionality. However, you can change logging parameters, if necessary, to better suit your needs.

Note: Before setting up the logging parameters, ensure that you understand the log4j utility. For detailed information about this utility, see <http://jakarta.apache.org/log4j>.

To use a log4j file inside the EAR:

1. Create new file in <INSTALL_DIR>/resources, for example log4jconfig.custom.xml.
2. Modify customer_overrides.properties used by your application server to add yfs.log4j.configuration=/resources/log4jconfig.custom.xml.
3. Run <INSTALL_DIR>/bin/deployer.sh -t resourcejar. This puts log4jconfig.custom.xml in resources.jar.
4. Rebuild and redeploy the EAR, following the instructions for your application server type, as described in the *Selling and Fulfillment Foundation: Installation Guide*.

To use a log4j file outside of the EAR:

1. Create new file in your classpath, for example /folder/log4jconfig.custom.xml.
2. Modify customer_overrides.properties used by your application server to add yfs.log4j.configuration=/folder/log4jconfig.custom.xml. Note that the "/" in front of the parameter is required.
3. Restart your application server.

| Property | Description |
|---|--|
| In the log4j configuration XML File | |
| <priority> subelement of the <root> element | <p>Specify the level of logging desired. Sterling Commerce recommends that you set the value of this attribute to ERROR.</p> <p>The following are valid values for logging levels:</p> <ul style="list-style-type: none">◆ ERRORDTL◆ ERROR◆ WARN◆ INFO◆ TIMER◆ SQLDEBUG◆ DEBUG◆ VERBOSE |

| Property | Description |
|--|---|
| <appender> subelement | <p>At the root level, this attribute specifies the associated name and class attribute. Choose a valid log4j appender class.</p> <p>Each subelement can also specify the layout of the message through the <layout> subelement and can filter for levels through the <filter> subelement.</p> <p>Instead of hardcoding the absolute path for the log file under the appender you plan to use, Sterling Commerce recommends that customers should also use a <code>LOG_DIRECTORY</code> parameter in the <code>log4jconfig.xml</code> file and invoke the JVM with a <code>-DLOG_DIRECTORY=<application_log_directory>/<logFileName></code> option.</p> |
| <param> subelement of the <appender> element | <p>This attribute specifies the associated name and value attribute. You can set the following variables using the <param> attribute:</p> <ul style="list-style-type: none"> ◆ <code>maxLogSize</code> - Specify the maximum number of write operations to be made to a log file (that is, this is not the memory size limit of the log file. The size of log file will depend on the size of each write operation.) By default, the value of this variable is set to <code>100000</code>. ◆ <code>rotateLogs</code> - Determines whether the log files should be split or not. If the value of this variable is set to <code>false</code>, the log files will not be split, and logger will keep writing in the same file. By default, the value of this variable is set to <code>true</code>. |

Masking Sensitive Information During Logging Using log4j

You can filter log messages when the log4j utility is used for logging. This helps prevent sensitive information, such as CVV2 codes, from being logged in the verbose log messages. By default, only the value of Secure Authentication Code is masked.

Selling and Fulfillment Foundation provides a custom log4j Layout and Filter. The custom Layout will first delegate to a normal PatternLayout to get a formatted message. When the formatted message is received, the custom Layout will filter the results based on a set of configurable regular expressions, before finally returning the fixed string. The custom Filter enables you to match the message against regular expressions and mask the message, if the message matches.

Notes:

- ◆ Filtering applies only to the messages logged using the log framework that is provided, which includes both YFCLogCategory and LogService. Messages logged through other methods, such as direct system.out or other log framework, will not be affected.
- ◆ If filtering is enabled, logging will be slower because it results in the execution of one or more regular expressions against every log message. This in turn may have a larger impact with the VERBOSE mode that is enabled for logging.

Using Custom Layout and Filter

Using the log4j utility, you can modify the log messages in the following places:

- ◆ Layout - Forms the actual message
- ◆ Appender - Writes the message

Selling and Fulfillment Foundation provides a custom layout, `SCIFilteredPatternLayout`, to handle message modification. You must change the layout class name in your custom logging configuration to `SCIFilteredPatternLayout`, for example:

```
<layout
class="com.sterlingcommerce.woodstock.util.frame.logex.SCIFilteredPatternLayout">
  <param name="ConversionPattern" value="%d:%-7p:%t: %-60m
  [%X{AppUserId}]: %-25c{1}%n"/>
  <param name="FilterSet" value="common-filter"/> <!-- Optional -->
</layout>
```

Selling and Fulfillment Foundation provides a custom filter, `SCIPatternFilter`, to filter out any messages that match certain regular expressions. You must change the filter class name in your custom logging configuration, for example:

```
<filter class="com.sterlingcommerce.woodstock.util.frame.logex.SCIPatternFilter" >
  <param name="FilterSet" value="suppress" /> <!-- Optional -->
</filter>
```

The following example calls an API with the following element in the input XML:

```
<Payment PaymentType="CREDIT_CARD" CreditCardNo="411kdiwbc6fj1111"
SecureAuthenticationCode="1234" MaxChargeLimit="100.00"/>
```

If the log level is set to `VERBOSE`, then the XML element will be logged as it appears above. Using the Log Filter, the following would be entered instead in the log file:

```
<Payment PaymentType="CREDIT_CARD" CreditCardNo="411kdiwbc6fj1111"
SecureAuthenticationCode="****" MaxChargeLimit="100.00"/>
```

Setting Up Regular Expressions

Selling and Fulfillment Foundation has introduced a property file, `logfilter.properties`, that can be used to set up named sets of regular expressions for both server-side masking and client-side masking of sensitive data. Each property is of the following form:

```
filterset.<name>.pattern.<num>=<pattern>
[optional]filterset.<name>.replace.<num>=<replace>
```

The pattern property is a Java-style regular expression, and defines the regular expression against which you want to match the message string. The replace property is optional, and defines what to replace the expression with. If the replace property is not defined, the default replacement string will be used. This property has no effect during a Filter. If the replacement string is invalid, the framework will catch the exception and print `***INVALID REPLACE VALUE***`. The original unfiltered messages are not logged.

Selling and Fulfillment Foundation provides the following properties that enable you to set the default FilterSet parameters for server-side masking:

- ◆ `default.filter.filterset=<filter_name>`
- ◆ `default.layout.filterset=<layout_name>`

Similarly, Selling and Fulfillment Foundation provides the following properties that enable you to set the default FilterSet parameters for client-side masking:

- ◆ `default.rcp.filter.filterset=<filter_name>`
- ◆ `default.rcp.layout.filterset=<layout_name>`

You can also define a common set of patterns across multiple filter sets, for example:

```
filterset.<name>.includes=<name1>,<name2>,...
```

Note: The following sample Log Filter Properties file is included for informational purposes only. You must set these properties in the `customer_overrides.properties` file, rather than editing them in `logfilter.properties.in` directly.

Sample Configuration for Log Filter Properties

The following sample configuration creates two sets of regular expression patterns, `suppress` and `common-filter`, and associates them with the default Filter and Layout configurations:

```
#Setting default Filter and Layout configurations for server-side masking
default.filter.filterset=suppress
default.layout.filterset=common-filter
```

```
#Setting default Filter and Layout configurations for client-side masking
#for Rich Client Platform based Applications
default.rcp.filter.filterset=rcpl
default.rcp.layout.filterset=rcp2
```

```
#The string "creditcardnumber" is suppressed in the log messages during
#logging. (?i) indicates case-insensitive matching.
```

```
filterset.suppress.pattern.2=Password\\s*\|=
#The string pattern "Password =" is suppressed in the log messages during
#logging.
```

```
filterset.common-filter.pattern.1=(Password|CVV|CreditCardNo)\\s*=\|s*([\" `
]).*?\|2
#The string pattern "Password =" is replaced with the string pattern
#mentioned in the replace property during logging.
```

```
filterset.common-filter.replace.1=$1=****
#The string pattern "Password =" is replaced with the string pattern
#"Password=*****" during logging.
```

```
filterset.suppress.includes=common-filter,<any_other_filter>
```

Notes:

- ◆ An empty pattern will be ignored. The number at the end does not matter, but it must be unique.
- ◆ If there is a loop in the dependencies, or if there is a preference to an invalid dependency, an exception is thrown. An exception is also thrown if a pattern is invalid. However, if the `logfilter.properties` file is missing, no exceptions are thrown, and nothing is filtered.

Enabling Different Properties for Individual Processes

It is possible to specify different properties for each process you are running. To do this, you must have a different `servers.properties` and `customer_overrides.properties` file for each process that you are running. In the start scripts for the process, set your `-DvendorFile=<your custom servers.properties>`. In your customer `servers.properties`, change the entry for `customer_overrides.properties` to point to your new `customer_overrides.properties`.

Configuring Cache for Catalog Search Index

Selling and Fulfillment Foundation uses Ehcache, a third-party caching framework, to manage memory that is used by the search index cache. With the SearchIndexCache configuration provided by Selling and Fulfillment Foundation, Least Recently Used (LRU) pre-search results for categories and filtered attributes in the catalog search index are cached to disk, thereby reducing the amount of memory usage.

Note: The information in this section describes the out-of-the-box Ehcache configuration values provided by Sterling Commerce. For detailed information about Ehcache functionality and how you can configure the elements and settings to enhance your cache for catalog search index, see <http://ehcache.org/> and <http://ehcache.org/documentation/EhcacheUserGuide-1.5.pdf>.

If you use catalog search index, you need to override and configure the out-of-the-box cache XML file provided by Selling and Fulfillment Foundation.

To configure the cache for catalog search index:

1. Copy the `<INSTALL_DIR>/resources/cache.xml` as `<INSTALL_DIR>/resources/cacheoverride.xml`.
2. Specify the location where the cached data overflow will be stored.

The disk location can be set in either of the following ways:

- ◆ Specify the path name in the `diskStore` element of `cacheoverride.xml`. For example:

```
<diskStore path="/<pathname>"/>
```
- ◆ If you need to override the disk storage path, specify the path name in the `sci.ehcache.disk.store.dir` property in the `yfs.properties` file. The disk storage path is automatically set to be a unique directory under the directory you specified. For additional information about the `sci.ehcache.disk.store.dir` property, see "[Property Files Reference](#)".

Note: The cached data can be stored either on a shared, central disk that is accessible from all servers, or it can be stored on a local server. If the cached data is stored on a shared disk, each application server requires its own disk store directory, which is named based on the system argument `jvmcacheid`. If this argument is not passed, the name of the disk store directory is the `ServerID` of the server.

Note: If the `sci.ehcache.disk.store.dir` property is set in the `yfs.properties` file, the value of this property will override the value of the disk store path set in `cacheoverride.xml`.

3. Selling and Fulfillment Foundation provides out-of-the-box cache configuration values for catalog search index in the SearchIndexCache section of `cacheoverride.xml`. You can modify the values of these elements to suit your business needs.

For example, the following table describes some of the elements that you can modify in the SearchIndexCache section of `cacheoverride.xml`. The `maxElementsInMemory` element described in this table defaults to 10000, which is a generous estimate. You can adjust this setting higher or lower for your specific requirements.

After setting this element, monitor your JVM and consider testing this in a staging environment before implementing it.

Note: Each pre-search result for a category or a filtered attribute is considered as an element in the cache.

| Element | Description |
|---------------------------|--|
| maxElementsInMemory | Required. Specifies the maximum number of elements that can be stored in memory. If you specify zero (0), there is no limit on the number of elements that can be stored in memory. By default, the value is set to 10000. |
| overflowToDisk | Required. Specifies whether or not elements are written to disk when the memory store has reached the maxElementsInMemory limit. Valid values are true or false. If set to true, the elements are written to disk when the memory store has reached the maxElementsInMemory limit. |
| maxElementsOnDisk | Required. Specifies the maximum number of elements that can be stored on disk. If you specify zero (0), there is no limit on the number of elements that can be stored on disk. By default, the value is set to 0. |
| eternal | Required. Specifies whether or not elements are eternal. By default, the value is set to true. Note: Do not change this default value. |
| diskPersistent | Optional. Specifies whether or not the disk store persists between restarts of the Java Virtual Machine (JVM). By default, the value is set to false. Note: Do not change this default value. |
| memoryStoreEvictionPolicy | Optional. Specifies the algorithm to be used for in-memory cache when the memory store reaches the maxElementsInMemory limit. Valid values: <ul style="list-style-type: none">◆ LRU (Least Recently Used)◆ FIFO (First In First Out)◆ LFU (Least Frequently Used) By default, the value is set to LRU. Note: Disk storage always uses the LFU algorithm. |
| diskSpoolBufferSizeMB | Optional. Specifies the size (in MB) to allocate the disk store for a spool buffer. Writes are made to this area and then asynchronously written to disk. By default, the size is set to 30 (MB). Note: Each spool buffer is used only by its cache. If you get OutOfMemory errors, consider lowering this value. To improve disk store performance, consider increasing this value. |

Note: After you create `cacheoverride.xml`, you must update and rebuild the `resources.jar` file. For additional information about building and deploying extensions to XML files in the `<INSTALL_DIR>/resources` directory, refer to the *Selling and Fulfillment Foundation: Customization Basics Guide*.

Property Files Reference

The following property files are described in this topic:

- ◆ `yfs.properties*`
- ◆ `dbclassCache.properties.*`
- ◆ `Sandbox.cfg`

Note: Sterling Commerce supports changes to the properties included in this chapter only. Changes to any other properties are not supported.

Making Changes to Properties

Do not directly edit or change the property files covered in this appendix, except `sandbox.cfg`. To make changes to the properties in these files, you must use the `customer_overrides.properties` file or `sandbox.cfg`. Sterling Commerce does not recommend that you modify or change any properties in files ending with `.in` because newer versions or patches of the product will overwrite your changes. Sterling Commerce also does not recommend that you change a property file that has a corresponding `.in` file because the `setupfiles` script will re-create the properties file again, thus causing you to lose your changes.

Setting the Database Connection Properties

Database connection properties are configured during installation. They are placed in the `<INSTALL_DIR>/properties/sandbox.cfg` file and are used to set values in the `jdbc.properties` file when the `setupfiles` utility is run.

The `DB_SCHEMA_OWNER` is set by the installer to the default schema of the user for whom the installation is being run. If you want to put objects in a schema other than the default schema for Oracle or for DB2, you must edit the `sandbox.cfg` file and change the `DB_SCHEMA_OWNER` variable to the desired value. The system will use the variable to decide what schema to work against.

Datasource Connection Pooling Property

In `customer_overrides.properties`, you can define an external datasource on your application server to manage connection pooling for your database as follows:

```
<property_file_descriptor>.<database>Pool.datasource=<datasource_name>
```

For example:

```
jdbcService.db2Pool.datasource=db2_transaction_pool
```

In this example:

- ◆ `jdbcService` is the property file descriptor.
- ◆ `db2` is the name of the database you are using with your application.
- ◆ `db2_transaction_pool` is the name of the datasource that you want to use to manage database connection pooling. You defined this in the application server.

Note: To use an application server datasource, you must also set the `-Dvendor` parameter to `oracle`, `jboss`, or `db2` in the application server script file. Refer to the *Selling and Fulfillment Foundation: Installation Guide* for more information about configuring datasource connection pooling.

yfs.properties*

The `yfs.properties*` files contain business-level properties. The properties are grouped in the following categories:

- ◆ [Agent](#)
- ◆ [Business Intelligence](#)
- ◆ [Catalog Management](#)
- ◆ [Database](#)
- ◆ [Exception Management](#)
- ◆ [Implementation](#)
- ◆ [Inventory Management](#)
- ◆ [JMS](#)
- ◆ [Context-Sensitive Help and Online Documentation Library](#)
- ◆ [Order Management](#)
- ◆ [Parcel Carrier Server](#)
- ◆ [Prints](#)
- ◆ [Security](#)
- ◆ [Service Definition Framework \(SDF\)](#)
- ◆ [Rich Client Platform \(RCP\)](#)
- ◆ [Sterling Sensitive Data Capture Server \(SSDCS\)](#)
- ◆ [System Management](#)
- ◆ [User Interface](#)
- ◆ [Warehouse Management](#)

The following table describes the properties that can be changed:.

| Property | Values | Description |
|--------------------------------|---------------------|--|
| Agent | | |
| yfs.agent.override.providerurl | Default is not set. | <p>AgentServer Override for provider url: set the property to the provider URL which will be used by ALL Agent servers. This property does not impact integration servers configured in the service builder. This property overrides the url configured for time triggered transactions in config.</p> <p>Example for WebLogic: yfs.agent.override.providerurl=t3://<host>:<port></p> <p>Example for WebSphere: yfs.agent.override.providerurl=corbaloc::<host>:<bootstrapport></p> <p>Example for JBoss: yfs.agent.override.providerurl=jnp://<ipaddress>:<port></p> |
| yfs.agent.override.icf | Default is not set. | <p>AgentServer Override for InitialContextFactory Name: set the property to the InitialContextFactory Name which will be used by ALL Agent servers. This property does not impact integration servers configured in the service builder. This property overrides the InitialContextFactory configured for time triggered transactions in config.</p> <p>Example for WebSphere: yfs.agent.override.icf=com.ibm.websphere.naming.WsnInitialContextFactory</p> <p>Example for WebLogic: yfs.agent.override.icf=weblogic.jndi.WLInitialContextFactory</p> <p>Example for JBoss: yfs.agent.override.icf=org.jnp.interfaces.NamingContextFactory</p> |
| yfs.agent.override.qcf | Default is not set. | <p>AgentServer Override for QueueConnectionFactory Name: set the property to the QueueConnectionFactory Name which will be used by ALL Agent servers. This property does not impact integration servers configured in the service builder. This property overrides the QueueConnectionFactory configured for time triggered transactions in config.</p> <p>Example: yfs.agent.override.qcf=<QueueConnectionFactory Name></p> |

| Property | Values | Description |
|----------------------------------|---------------------|--|
| yfs.agent.override.retryCount | Default is not set. | <p>AgentServer Override for RetryCount: set the property to the JMS RetryCount to be used by ALL Agent Servers. This property does not impact integration servers configured in the service builder.</p> <p>Example:</p> <p>yfs.agent.override.retryCount=<Number of Retries></p> |
| yfs.agent.override.retryInterval | Default is not set. | <p>AgentServer Override for RetryInterval: set the property to the JMS RetryInterval to be used by ALL Agent Servers. This property does not impact integration servers configured in the service builder. This interval is specified in milliseconds.</p> <p>Example:</p> <p>yfs.agent.override.retryInterval=<RetryInterval in Milliseconds></p> |

| Property | Values | Description |
|---------------------------------|-----------------------|--|
| yfs.agent.backup.providerurl | Default is not set. | <p>AgentServer Backup for JMS. Set these three properties to the JMSprovider URL InitialContextFactory Name and QueueConnectionFactory Name which will be used by ALL Agent servers as a Backup if the primary JMSServer becomes unavailable, the QueueName used by the Agent Server on the backup JMSServer will be the same as the one configured on the primary JMSServer.</p> <p>These properties will be used only after a successful start of the Agent Server using the primary JMSServer configuration. If the AgentServer Override properties are specified, they are used as the primary JMSServer properties. The backup JMSServer will be used by the Agent Server only if all three properties are specified.</p> <p>These properties do not impact integration servers configured in the service builder.</p> <p>Examples for WebSphere:</p> <pre>yfs.agent.backup.providerurl=corbaloc:<host>:<boostrapport></pre> <pre>yfs.agent.backup.icf=com.ibm.websphere.naming.WsnInitialContextFactory</pre> <p>Examples for WebLogic:</p> <pre>yfs.agent.backup.providerurl=t3://<host>:<port></pre> <pre>yfs.agent.backup.icf=weblogic.jndi.WLInitialContextFactory</pre> <p>Examples for JBoss:</p> <pre>yfs.agent.backup.providerurl=jnp://<ipaddress>:<port></pre> <pre>yfs.agent.backup.icf=org.jnp.interfaces.NamingContextFactory</pre> |
| yfs.agent.backup.icf | Default is not set. | |
| yfs.agent.backup.qcf | Default is not set. | |
| yfs.agent.backup.retryCount | Default = 0 | <p>Number of times to retry a failed JMS connection</p> <p>Example: yfs.agent.backup.retryCount=3</p> |
| yfs.agent.backup.retryInterval | Default = 0 | <p>Number of milliseconds to wait between retries</p> <p>Example: yfs.agent.backup.retryInterval=3</p> |
| yfs.agent.override.auth.enabled | Valid values = Y or N | <p>The agent override for the JMS Security parameter values specified in the Agent criteria. If set to Y, you must also include values for the following two properties (userid and password).</p> <p>Example:</p> <pre>yfs.agent.override.auth.enabled=Y</pre> |

| Property | Values | Description |
|--|---------------------|---|
| yfs.agent.override.auth.use rid yfs.agent.override.auth.pas sword | Default is not set. | If yfs.agent.override.auth.enabled is set to Y, these properties (userid and password) must be present, otherwise an error is thrown. |
| yfs.agentserver.queryTime out | | Set this property to specify the Query Time out for Agents. This is the global property applicable for all agents. It is possible for individual agents to override this property by specifying their own query timeouts. For example, the User Activity Audit Purge Agent can specify a property such as USERACTAUDITPRG.queryTimeout. This will override this global property. Set to zero means unlimited. Example: yfs.agentserver.queryTimeout= |
| Business Intelligence | | |
| analytics.portal.url | | Powerplay Launch URL. Replace the <machine> with the Cognos installation machine name/ip address. Do not change anything else in the URL, else Business Intelligence access will fail. Example: http://<machine>/cognos/cgi-bin/login.cgi?signon=#USERID#&password=#PASSWORD#&return_url=upfcgi.exe |
| analytics.reportnet.url | | Reportnet Analytics Launch URL. Replace the <machine> with the Cognos installation machine name/ip address. Do not change anything else in the URL, else Cognos access will fail Example: http://<machine>/crn/cgi-bin/cognos.cgi |
| analytics.namespace | | Business Intelligence namespace. Indicates the namespace that has been configured in COGNOS Business Intelligence authentication, against which the users would be authenticated. Example: <smcfs> |
| yfs.analytics.checkCached Reports | | Set this to Y to view the cached reports from Yantra Analytics Console. Example: yfs.analytics.checkCachedReports=N |

| Property | Values | Description |
|-----------------------------------|--|--|
| Catalog Management | | |
| yfs.searchIndex.RootDirectory | <root path of catalog search index file> | This property must be set if you use catalog search. It contains the root path to the directory where the catalog search index file will be written. Example: yfs.searchIndex.RootDirectory=<root path of catalog search index file> |
| yfs.searchIndex.maxNumberOfErrors | Default = 3 | This property determines the number of errors that the Catalog Index Build agent should encounter before it stops trying to process a search index trigger. By default, if the search index trigger encounters 3 errors, the trigger will be marked as a failure, and it will not be picked up again by the agent. |
| yfs.searchIndex.maxRetryCount | Default = 3 | This property determines the number of times the agent will sleep waiting for the process to finish in the event that the last message is processed before all other messages have completed processing. The <code>yfs.searchIndex.sleepCount</code> property determines the amount of time the agent sleeps between retry attempts. |
| yfs.searchIndex.sleepCount | Default = 300,000 | This property determines the number of milliseconds the index building agent waits before trying to merge individual message files into the final index file. |
| sci.ehcache.disk.store.dir | <root path of cached data> | This property contains the root path to the directory where cached objects will be written. For example: sci.ehcache.disk.store.dir=<root path of cached data> Note: If this property is set in the <code>yfs.properties</code> file, the value of this property will override the value of the disk store path set in the cache configuration XML file. |
| Database | | |
| yfs.db.compression.class | Valid values = <class name> | Enter the custom class which provides the implementation logic for the data compression for columns that support compression. |

| Property | Values | Description |
|---|--|--|
| yfs.dblogin.yantraschema.name | Valid values = <schema name> Default is not set. | The database schema for the Application installation (if different from the userid being used) Example: yfs.dblogin.yantraschema.name=<SchemaName> |
| yfs.cursor.sharing.mode.dcm | Default = FORCE | Note: This property is required only when the database is Oracle and you are integrating with Yantra DCS. Set this to the current value of cursor_sharing parameter in your database. You can find it in v\$parameter table of Oracle. Example: yfs.cursor.sharing.mode.dcm=FORCE |
| yfs.db.textsearch | Valid values = Y or N Default = N | Setting this property will decide whether Text Search is to be enabled. Example: yfs.db.textsearch=N |
| yfs.db.textsearch.oracle.contexttype | Valid values = CTXCAT or CONTEXT Default = CTXCAT | Set this property to determine the type of text index (ctxcat or context) in Oracle. Example: yfs.db.textsearch.oracle.contexttype=ctxcat |
| SUFFIX_KEY_WITH_INSTANCE_NO | Valid values = Y or N Default = N | Set this property to insert instance number into primary keys generated. Example: SUFFIX_KEY_WITH_INSTANCE_NO=N |
| Exception Management | | |
| yfs.exception.disable.uniqueExceptionId | Valid values = Y or N Default = N | When this property is turned off (set to N), the system generates a unique exception ID based on the IP address and random numbers when an error is thrown. The ID appears in the output XML of the error. If you do not want to generate unique exception IDs, set this property to Y. Example: yfs.exception.disable.uniqueExceptionId=N |
| yfs.onerror.raisealert | Valid values = Y or N Default = Y | When this property is turned on (set to Y), errors encountered are directed to the alert console. Errors are grouped by certain criteria. Not every error is a separate entry in the alert console. Example: yfs.onerror.raisealert=Y |

| Property | Values | Description |
|---|---|---|
| yfs.onerror.raisealert.logging.interval | Valid values = day or hour Default = day | This property is to set the logging interval. If it is set to day, errors are grouped on daily basis. Otherwise, they are grouped on an hourly basis. If an error is repeated within a logging interval, it is not directed multiple times to alert console. Instead, the alert entry shows an increased occurrence count. Example: yfs.onerror.raisealert.logging.interval=day |
| yfs.onerror.raisealert.expiration.days | Number of days Default = 7 A value of 0 means the alert will never be closed by this agent. | This property controls how many days of inactivity are required before the alert is eligible to be automatically closed by the Inbox Purge Agent. Example: yfs.onerror.raisealert.expiration.days=7 |
| yfs.onerror.raisealert.queuekey | <QUEUE_KEY> Default = DEFAULT | Set this to the QUEUE_KEY of the Queue the alert should be assigned to. Example: yfs.onerror.raisealert.queuekey= |
| yfs.exception.display.uniqueExceptionID=Description | | This property tells the application where to add the unique exception id, either on error description (default behavior) or in the error XML as a separate element. |
| yfs.exception.disable.uniqueException=N | | If this property is set to Y, the system does not append a unique numeric ID to the error description of an error. The default value is N, which means that a unique numeric ID gets appended to the error description. |
| Implementation | | |
| yfs.context.namespace | Default is not set | Prefix to add the JNDI name when performing an EJB lookup in the client API library. If not set, then no prefix is used. |

| Property | Values | Description |
|-------------------------|--|--|
| yfs.api.history.disable | Valid values = true or false Default = false | <p>This property controls access to history tables. If the value of this flag is set to "true", access to history tables is suppressed and the select, list, and count operations on history tables will return no data.</p> <ul style="list-style-type: none"> ◆ A select operation returns null. ◆ A list operation returns an empty list. ◆ A count operation returns a zero count. <p>Also, if update, delete, and insert operations are attempted on history tables, the system will throw an exception.</p> <p>This property enables applications to suppress history so that the system can be started after upgrading transaction data, even before history upgrades are complete.</p> <p>Note: When the value of this flag is changed while the application is running, the behavior of the application also changes on the fly.</p> <p>Example: yfs.api.history.disable = true</p> |
| yfs.comsupport | Valid values = Y or N Default = Y | <p>Indicates whether or not your system supports COM. If COM is enabled, you can configure actions to call COM objects.</p> <p>Example: yfs.comsupport=Y</p> |
| yfs.purge.path | Default is &APP_DIR;/logs | <p>This property must be set for the purge programs to run. It contains the absolute path to the directory where purge logs will be written.</p> <p>Example: yfs.purge.path=&APP_DIR;/logs</p> |
| log4j.configuration | <Property Name> Default = /resources/log4jconfig.xml | <p>Property to handle logging. This property points to the location of the log4j configuration xml file.</p> <p>Example: log4j.configuration=/resources/log4jconfig.xml</p> |

| Property | Values | Description |
|-----------------------------------|---|---|
| CaseInsensitiveSearch.Mode | Valid Values = DISABLED, ENABLED, or MIXED Default = ENABLED | <p>This property specifies the following configuration modes to enable or disable case insensitive search across entities.</p> <ul style="list-style-type: none"> ◆ Disabled: Search is case-insensitive. Search is done on the original column only. ◆ Enabled: Search is case insensitive. Search is done on the shadow column, which is generated once this attribute is set. ◆ Mixed: Search is done on both the original and shadow columns. This mode is useful only during data migration (while populating the shadow column). <p>Note: Mixed mode might have a negative impact on performance, hence it is recommended to not use mixed mode.</p> <p>Though this mode is globally applicable across entities, it can be overridden at a particular entity or column level.</p> <p>To enable case insensitivity at an entity level, use the format:</p> <pre><ENTITY_NAME>.CaseInsensitiveSearch.Mode=ENABLED</pre> <p>Example:</p> <pre>YFS_INBOX.CaseInsensitiveSearch.Mode=ENABLED</pre> <p>To enable case insensitivity at a column level, use the format:</p> <pre><ENTITY_NAME>.<COLUMN_NAME>.CaseInsensitiveSearch.Mode=ENABLED</pre> <p>Example:</p> <pre>YFS_INBOX.ALERT_TYPE.CaseInsensitiveSearch.Mode=ENABLED</pre> |
| yfs.install.localecode | <locale code> Set to en_US_EST when Selling and Fulfillment Foundation is delivered; If changes are necessary, this must be changed by the customer. | <p>Installation locale code. This localecode has to match the underlying OS default timezone where the database is installed to avoid timezone calculation errors.</p> <p>Example:</p> <pre>yfs.install.localecode=en_US_EST</pre> |
| yfs.install.displaydoublequantity | Valid values = Y or N Default = Y | <p>This property should be set to "Y" if you want to support fractional quantities for attributes which belong to QUANTITY datatype.</p> <p>Example:</p> <pre>yfs.install.displaydoublequantity=Y</pre> |

| Property | Values | Description |
|----------------------------------|---------------------------------------|---|
| yfs.file.encoding | <Property name> Default = UTF-8 | This property controls the encoding of the files, like API template, theme XMLs, exception template (except e-mail templates which is controlled by yfs.email.template.encoding). Example: yfs.file.encoding=UTF-8 |
| yantra.app.maxrecords | <number of records> Default = 5000 | This property sets the default number of records returned by Application list APIs. Increase the application server JVM heap settings if these parameters are increased. Change will affect search limits for all users. Example: yantra.app.maxrecords=5000 |
| yfs.app.identifyconnection | Valid values = Y or N Default = N | Set this property to "Y" to enable the application to set contextual information (Agent, API name) on the connection. The information stamped on the connection can be viewed on the database connection by the tools provided by the database vendors. This allows mapping of the connection in the database to its origin on the application side. Note: This is available only for Oracle and DB2. Example: yfs.yfs.app.identifyconnection=Y |
| yfs.uidev.refreshResources | Valid values = Y or N Default = N | This property can be set to "Y" when developing and customizing the Application Console UI. When set to "Y", refresh actions will be available within the Resource Hierarchy tree. These actions can be used to refresh the resources used in the Console UI Framework without having to restart the application server. If this property is not set to "Y", then the actions will be disabled, and it will be necessary to restart the application server whenever any resource is changed within the Applications Manager. IMPORTANT: This property should NOT be set to "Y" for live production systems because the refresh only works for a single user development environment. Example: yfs.uidev.refreshResources=N |
| yantra.document.isnamespaceaware | Valid values = Y or N Default = N | This property should be set to Y to handle namespaces in XML. Example: yantra.document.isnamespaceaware=N |

| Property | Values | Description |
|--|--|--|
| yantra.document.suppress.huge.doc.alert | Valid values = true or false Default = false | Set this property to enable or disable printing a document that contains more than one million nodes. If the value of this property is set to "true", the document will not be printed, and instead, a "Please disable/set yantra.document.suppress.huge.doc.alerts to False to print the offending document with more than one million nodes" warning message will be displayed. If set to "false", the application will retain the default behavior. |
| com.yantra.ycp.em.server.taskpollingtime | <number of seconds> Default = 60 | Set this property (in seconds) to indicate the interval at which tasks need to be polled for mobile operators. Example: com.yantra.ycp.em.server.taskpollingtime=60 |
| OverrideTransitTime | Valid values = Y or N Default = N | If both Request Delivery date and Request Ship date are passed, setting this property will compute the transit time as difference between these two dates, and no further transit time computation will be required. Example: OverrideTransitTime=N |
| yfs.install.applyshipdateoptimization | Valid values = Y or N | Set this property to "N" if allocation should not optimize releases based on ReqShipDate. Example: yfs.install.applyshipdateoptimization=Y |
| yfs.audit.user.session.activity | Valid values = Y or N Default = Y (audit enabled) | Set this property to enable auditing and disable auditing. Example: yfs.audit.user.session.activity=Y |
| yfs.bundleProperties.multipleLoad=Y | Default: locale is en_US | When this property is set to N, the system will not create multiple entries for bundle literals for different locales. When trying to load literals for a given locale, if the properties file is not found for the locale, the system will use the default locale for string translations. |
| yfs.logall | Valid values = Y or N Default =N | Setting this property enables and disables verbose logging. Example: yfs.logall=N |

| Property | Values | Description |
|---------------------------------|--------------------------------------|--|
| Inventory Management | | |
| yfs.inventory.sortandlock | Valid values = Y or N | Setting this property to Y will prevent dead locking on YFS_Inventory_Item table during order creation and when synchronizing changes from a shipment back to the order. This parameter is only used if the order contains a logical kit, or when a shipment contains shipment lines from different orders. Setting this property to Y will result into holding lock on YFS_Inventory_Item table for longer period. So, this may degrade overall system performance. Because of this reason, you should set this property to Y only if you are getting excessive dead locks on YFS_Inventory_Item table. Example: yfs.inventory.sortandlock= |
| yfs.hotsku.useHotSKUFeature | Valid values = Y or N Default = N | Set this property to "Y" to use the Hot SKU feature. For more information about the Hot SKU feature, please see <i>Selling and Fulfillment Foundation: Product Concepts Guide</i> and the <i>Selling and Fulfillment Foundation: Performance Management Guide</i> . Example: yfs.hotsku.useHotSKUFeature=N |
| yfs.hotsku.useTimeOutLocking | Valid values = Y or N Default = N | Set this property to "Y" to use locking timeout while trying to obtain a lock for an inventory item when the Hot SKU feature is enabled. The timeout period will be based on the yfs.hotsku.secondsToClassifyAsAbnormalTime property (round up to integer). Example: yfs.hotsku.useTimeOutLocking=N |
| JMS | | |
| yfs.jms.session.disable.pooling | Valid values = Y, N Default = N | To disable JMS Session pooling, set this property to Y. Sterling recommends setting to N for performance reasons. Example: yfs.jms.session.disable.pooling=N |

| Property | Values | Description |
|--|--|---|
| yantra.jms.receive.timeout | Valid values = <number of milliseconds> Default = 1800000 milliseconds (30 minutes) | Specifies the JMS receive timeout value (in milliseconds). The JMS Receiver receives the next message that arrives within the specified timeout interval. The receive call blocks until a message arrives, the timeout expires, or this message consumer is closed. A timeout of zero never expires, and the call blocks indefinitely. Example: yantra.jms.receive.timeout=180000 Note: If the default JMS server of Websphere is used, then the receive timeout interval must be set to 60000 milliseconds (1 minute). Example: yantra.jms.receive.timeout=60000 |
| yfs.flow.override.auth.enabled yfs.flow.override.auth.userid yfs.flow.override.auth.password | Valid values = Y or N | The flow overrides to override the JMS Security parameter values specified in the service definition framework. If this value is set to Y then the other two properties at left (userid and password) must be present, otherwise an error is thrown. Examples: yfs.flow.override.auth.enabled=Y yfs.flow.override.auth.userid=Y yfs.flow.override.auth.password=Y |
| yfs.jms.session.disable.pooling | Valid values = Y or N Default = N | To disable JMS Session pooling, set this property to Y. By default this property is set to N. Yantra recommends N value to this property for better performance reasons. Example: yfs.jms.session.disable.pooling=N |
| yfs.restrict.flowmessageid.lengthto24 | Valid values = Y or N Default = N | To restrict the length of correlating attributes, such as MESSAGEID and JMS-CORRELATION-ID, to 24 characters, set this property to Y. Example: yfs.restrict.flowmessageid.lengthto24=N |
| sci.queuebasedsecurity.userid | Valid values = <USER_ID> | Set this property for queue-based security. Example: sci.queuebasedsecurity.userid=<user_id configured in the APPLICATION_SERVER and assigned to the queue> |
| sci.queuebasedsecurity.password | Valid values = <PASSWORD> | Set this property for queue-based security. Example: sci.queuebasedsecurity.password=<password for the user_id as configured in the APPLICATION_SERVER> |

| Property | Values | Description |
|--|---|--|
| Context-Sensitive Help and Online Documentation Library | | |
| yfs.urlforhelp.path | </link to Context-Sensitive Help/> Default = /wwhelp/wwhimpl/common/html/wwhelp.htm | This controls where the link to the Context-Sensitive Help will go. The value must begin and end in a forward slash (/) to ensure proper behavior. Example: yfs.urlforhelp.path=/wwhelp/wwhimpl/common/html/wwhelp.htm Note: Context-Sensitive Help is not the same as the Product Documentation Libraries , also described in this guide. |
| yfs.heightforhelp.path | <height in pixels> Default = 590 | This controls height for the Context-Sensitive Help. Example: yfs.heightforhelp.path=390 |
| yfs.widthforhelp.path | <width in pixels> Default = 715 | This controls width for the Context-Sensitive Help. Example: yfs.widthforhelp.path=515 |
| yfs.onlinehelp.path | </Context-Sensitive Help path/> Default = /<appname>docs/yfsccommon/online_help/ | This controls where the link to the Context-Sensitive Help will go. The value must begin and end in a forward slash (/) to ensure proper behavior. It must be a simple path; a full URL will not work. Example: yfs.onlinehelp.path=<appname>docs/yfsccommon/online_help/ |
| yfs.onlinehelp.path.overrideforlocale.<lang_country code> | Valid values=</Context-Sensitive Help path/ lang_country code> Default=/<appname>docs/yfsccommon/online_help/en_US | This property enables you to specify the path for a locale-specific Context-Sensitive Help. This property overrides the yfs.onlinehelp.path property. The default locale used for Context-Sensitive Help is US English (en_US). If this property is not defined, the path specified in the yfs.onlinehelp.path property is considered, and is suffixed with the default language and country code, which is, en_US. Example: yfs.onlinehelp.path.overrideforlocale.jp_JP=/yfantrdocs/yfsccommon/online_help/jp_JP |

| Property | Values | Description |
|------------------------------------|---|--|
| yfs.sync.jms.request.timetolive. | | JMS messages. Use this property to set the timetolive value for Synchronous JMS Messages. Note that, this property is a global level override and is applicable for all the services using Synchronous JMS. WARNING: If this property value is set beyond the responseTimeout value, it may result in unprocessed request messages in the reply/response queue. By default, in order to clean up the un-processed messages from the reply/response queue, the timeToLive for request messages is set to the responseTimeout value of the receiver. Example: yfs.sync.jms.request.timetolive=<responseTimeout valueof the receiver> |
| plt.active.doc.url | Valid values: LOCAL or ONLINE Default: ONLINE | Flag to identify Active Help URL. |
| plt.online.doc.url | | Absolute URL for the Sterling-hosted web page that contains the Selling and Fulfillment Foundation Online Documentation Library help content. |
| plt.local.doc.url | | Relative URL for the local instance of the Selling and Fulfillment Foundation Local Documentation Library help content. |
| Order Management | | |
| yfs.transferPreparation.minTimeReq | <number of hours> Supported maximum value = 24 HRs Default = 0 | Set this property with the minimum time (Hours) required for transfer preparation. This property is used by all Promising and Scheduling APIs. Example: yfs.transferPreparation.minTimeReq=0 |
| Parcel Carrier Server | | |
| ycs.airborne.server.url | Valid values = <server url> Default = https://eCommerce.airborne.com/A piLandingTest.asp | Airborne logon parameter. Example: ycs.airborne.server.url=https://eCommerce.airborne.com/ApiLandingTest.asp |
| ycs.airborne.server.UserID | Valid values = <userid> | Airborne logon parameter. Example: ycs.airborne.server.UserID=<your airborne server userid> |
| ycs.airborne.server.Password | Valid values = <password> | Airborne logon parameter. Example: ycs.airborne.server.Password=<your airborneserver password> |

| Property | Values | Description |
|---|---|---|
| ycs.airborne.LabelPrint.X-coordinate | Valid values = <X-coordinate> Default = 0.1 | Airborne print parameter. Example: ycs.airborne.LabelPrint.X-coordinate=0.1 |
| ycs.airborne.LabelPrint.Y-coordinate | Valid values = <Y-coordinate> Default = 0.1 | Airborne print parameter. Example: ycs.airborne.LabelPrint.Y-coordinate=0.1 |
| ycs.airborne.LabelPrint.Width | Valid values = <LabelPrint.Width> Default = 8.5 | Airborne print parameter. Example: ycs.airborne.LabelPrint.Width=8.5 |
| ycs.airborne.LabelPrint.Height | Valid values = <LabelPrint.Height> Default = 6.5 | Airborne print parameter. Example: ycs.airborne.LabelPrint.Height=6.5 |
| ycs.airborne.LabelPrint.Dump_Switch | Valid values = yes or no Default = yes | Airborne print parameter. Example: ycs.airborne.LabelPrint.Dump_Switch=yes |
| ycs.connectship.server.url | <IP address> | Change the IP address according to your ConnectShip integration IP. Example: ycs.connectship.server.url=http://127.0.0.1:100/Logistics/XML_Processor/Server/XMLProcDLL.asp |
| ycs.fedex.powershipserver.ipaddress ycs.fedex.powershipserver.portno | <IP address> <port> | Change the IP address according to your Powership Server IP. Examples: ycs.fedex.powershipserver.ipaddress=127.0.0.1 ycs.fedex.powershipserver.portno=2000 |
| yfs.log.logger | <absolute path to message log dir> | This property must be set if you use the default message handling implementation. It contains the absolute path to the directory where message logs will be written. Example: yfs.log.logger= |
| ycs.log.directory ycs.xmlDump.directory | <full path to log dir> | This property must be set if you use the default message handling implementation. It contains the absolute path to the directory where message logs will be written. Examples: ycs.log.directory=<full path of log directory> ycs.xmlDump.directory=<full path of log directory> |

| Property | Values | Description |
|--|---|---|
| ycs.pierbridge.db.url | <IP address> <port> | Change the IP address and port number according to the IP and Port of your Pierbridge server database. Example: ycs.pierbridge.db.url=jdbc:sqlserver://10.11.20.133:1215;databaseName=Pierbridge Shipment Server |
| ycs.pierbridge.db.username | <userid> | This is the user identification required to log in to the Pierbridge server database. Example: ycs.pierbridge.db.username=yantra |
| ycs.pierbridge.db.password | <password> | This is the password required to log in to the Pierbridge server database. Example: ycs.pierbridge.db.password=yantra1 |
| ycs.pierbridge.server.url | <IP address> | Change the IP address according to your Pierbridge integration IP. Example: ycs.pierbridge.server.url=http://10.11.20.133//pierbridge%20shipping/services/xmlservice.aspx |
| ycs.pierbridge.default.rate.request.user | <userid> | This is the user identification required to log in to the Pierbridge shipment server. This user identification is required when the getFreightCharge API is invoked in the Sterling Warehouse Management System. Example: ycs.pierbridge.default.rate.request.user=pierbridge |
| ycs.timer.switch | Valid values = yes or no | This property is to set to enable and disable the Timer from logging. Example: ycs.timer.switch=yes |
| ycs.log.size.maxallowed | <max log file size in bytes> Default = 1000000 | This property is only used if you use the default message handling implementation. It specifies the maximum allowed log file size in bytes. A new active log file is created whenever the currently active log file reaches this size. Example: ycs.log.size.maxallowed=1000000 |

| Property | Values | Description |
|------------------------------|--|---|
| ycs.log.logger | <class name> | Default class to handle messages output by PureEcommerce. The default implementation will write these messages to log files. Refer to the Programming Guide for how to write your own implementation for this class. Example: ycs.log.logger=com.yantra.ycs.util.YCSFileLogger |
| ycs.purge.path | <full path of log directory> | This property must be set for the purge programs to run. Example: ycs.purge.path=<full path of log directory> |
| ycs.purge.days | <number of days> | This property must be set for the purge programs to run. Example: ycs.purge.days=15 |
| ycs.weight_tolerance_percent | <percent> Default = 5 | If a label is printed before it is actually manifested, weight printed on the label may not exactly match the actual weight. Parameter below specifies the tolerance percent beyond which the label has to be reprinted. Example: ycs.weight_tolerance_percent=5 |
| proxySet=false | Valid values = true or false Default = false If set to true, the following details are required: https.proxyHost https.proxyUser https.proxyPassword https.proxyPort | Proxy settings for https connections. Example: proxySet=false |
| Prints | | |
| yfs.software.tcpip.sockets | Valid values = Y or N Default = N | Set this property to Y to enable printing via TCP/IP Socket Interface. Set it to N to enable printing via the File Interface (file "drop" to a shared network drive). Note: Although the default value is set to N for the sake of backward compatibility, our recommendation is to set this flag to Y. Example: yfs.software.tcpip.sockets=N |

| Property | Values | Description |
|---------------------------------|--|--|
| yfs.loftware.tcpip.sockets.mode | Valid values = WAIT or NOWAIT Default = NOWAIT | Set this property to WAIT to change the Software print server's mode to WAIT mode Example: yfs.loftware.tcpip.sockets.mode=NOWAIT |
| Security | | |
| api.security.enabled | Valid values = Y or N Default = Y | Set this property to secure access to APIs. If enabled, an authorization check is performed on a user's access to an API when the user calls that API. |
| api.security.mode | Valid values = STRICT, LAX, or DEBUG Default = STRICT | <p>STRICT: If any validation fails, throw an exception. This is appropriate for production systems, if all permissions are configured properly.</p> <p>LAX: Filter out and log invalid input, but continue processing. The filtering allows the system to mostly work despite incorrect input or output, while the logging helps to identify places that need change. LAX can be useful during initial development and testing.</p> <p>DEBUG: Log invalid input and output, but do not filter anything or throw exceptions. This is appropriate only during initial development, to identify the permissions required by various processes.</p> <p>Note: If you do not specify a security mode, then the system defaults to STRICT.</p> <p>Example: api.security.override.createorder.mode=DEBUG</p> <p>This example sets access for the CreateOrder API to DEBUG.</p> |
| api.security.token.enabled | Valid values = Y or N Default = Y | If api.security.token.enabled = Y, the login API returns a special UserToken security token attribute upon successful authentication. The api.security.token.timeout property controls how long this token is active. |

| Property | Values | Description |
|---------------------------------------|--------------------------------------|--|
| api.security.console.enabled | Valid values = Y or N Default = N | <p>If api.security.console.enabled = Y, API security, in addition to the built-in security, is used for the JSP console.</p> <p>Note: Enabling this property may require that you relax other security settings or take additional steps that are described in the following article:</p> <ul style="list-style-type: none"> ◆ Go to the link https://customer.sterlingcommerce.com/group/sterling/support_center ◆ Log in, and in the Search Knowledgebase, enter “HTG2798”. ◆ The article will be displayed as a selectable item. |
| api.security.smc.enabled | Valid values = Y or N Default = N | <p>If api.security.smc.enabled = Y, API security, in addition to the built-in security, is used for the Applications Manager and the system monitor console.</p> <p>Note: Enabling this property may require that you relax other security settings or take additional steps that are described in the following article:</p> <ul style="list-style-type: none"> ◆ Go to the link https://customer.sterlingcommerce.com/group/sterling/support_center ◆ Log in, and in the Search Knowledgebase, enter “HTG2798”. ◆ The article will be displayed as a selectable item. |
| api.outputDBPasswords | Valid values = Y or N Default = N | If api.outputDBPasswords = true, passwords are returned in the output of the getDBPoolList and getDBConnParams APIs. |
| dsg.api.disable | Valid values = <api_name> | <p>Set this property to disable the new data access policy functionality for specific APIs.</p> <p>Example:</p> <p>dsg.api.disable=<api_name, api_name, api_name></p> |
| interopservlet.security.enabled | Valid values = Y or N Default = Y | Enables application server to authenticate a user, using token-based or container-based authentication. |
| interopservlet.auth.container.enabled | Valid values = Y or N Default = N | <p>Set this property to “true” if you want the application server to authenticate a user by checking whether the user ID matches the requested user ID. If this property is set to “false”, container-based authentication is disabled.</p> <p>Example:</p> <p>interopservlet.auth.container.enabled = false</p> |

| Property | Values | Description |
|--|---|---|
| interopservlet.auth.token.enabled | Valid values = Y or N Default = Y | Setting this property to “true” validates the user token supplied as a parameter on the request. When enabled, this also allows access to the login API, which is what supplies the user token. If this property is set to “false”, token-based authentication is disabled. Example: interopservlet.auth.token.enabled = true |
| interopservlet.auth.userPassword.enabled | Valid values = Y or N Default = Y | Set this property to “true” if you want the user ID and password to be passed as parameters to the servlet, instead of using the typical login API/token approach. Example: interopservlet.auth.userPassword.enabled = true |
| userauthfilter.enabled | Valid values = yes or no Default = Y | Sets a servlet filter that ensures authenticated user access to everything under web root, except for login pages. |
| yfs.login.singlesignon.class | <class name> | The class that handles Single Signon. Refer to the javadocs for the com.yantra.ycp.japi.util.YCPSSOManager interface for information about how to write your own implementation for this class. |
| yfs.security.singlesignon.enabled | Valid values = Y or N Default = Y | If this property is set to Y, the single sign on class is called. |
| yfs.login.singlesignon.checkuser | Valid values = Y or N Default = N | If this property is set to Y, each request is validated against the singleSignOn Server for the user authentication. If the property is set to N, user authentication against the singleSignOn Server is done only when the session times out. |
| yfs.security.authenticator | <class name> Default is not set. | The class that will be invoked for user authentication. Uncomment and change this only if you <u>do not</u> want to use application authentication. Refer to the javadocs for the YFSAuthenticator interface for information about how to write your own implementation for this class. If you want to use the default implementation for LDAP authentication, uncomment the property and set it to com.yantra.yfs.util.YFSLdapAuthenticator. Example: yfs.security.authenticator= |

| Property | Values | Description |
|---------------------------|---------------------|---|
| yfs.security.ldap.factory | <class name> | The LDAP context factory classname as specified in your LDAP Server configuration. Example: yfs.security.ldap.factory=com.sun.jndi.ldap.LdapCtxFactory |
| yfs.security.ldap.url | <url> | The URL for accessing your LDAP Server as specified in your LDAP Server configuration. Example: yfs.security.ldap.url=ldap://<ldapservername>:<portnum> |
| yfs.security.ldap.ou | Default is not set. | The value specified for the organizational unit in your LDAP Server configuration. Example: yfs.security.ldap.ou= |
| yfs.security.ldap.o | Default is not set. | The value specified for the organization in your LDAP Server configuration. Example: yfs.security.ldap.o= |
| yfs.encrypter.class | <class name> | The class that handles encryption and decryption of credit card numbers. Refer to the javadocs for the YCPDecrypter interface for information about how to write your own decrypter class. If this property is not specified, then no encryption will be performed. Sterling Commerce provides an application, the Sterling Sensitive Data Capture Server, that captures and tokenizes credit card numbers and store value card numbers. Sterling Commerce recommends that you review the <i>Selling and Fulfillment Foundation: Secure Deployment Guide</i> for Sterling's approach to meeting PCI DSS and PA-DSS requirements. |

| Property | Values | Description |
|---|--|---|
| yfs.propertyencrypter.class | Valid values = <class name> Default is not set. | <p>This class will be used for encrypting and decrypting properties specified in yfs.properties, yiclient.properties and management.properties files. All properties which end with ".encrypted" are automatically decrypted using this class at runtime. Use this property to encrypt critical data like user/password.</p> <p>The yfs.propertyencrypter.class is deprecated and no longer used for property encryption. You can now use security.property.encrypter.class for property encryption. For more information about encryption through property files, refer to the <i>Selling and Fulfillment Foundation: Extending Transactions Guide</i>.</p> <p>Example: yfs.agent.override.auth.password=<password> could be specified as: yfs.agent.override.auth.password.encrypted=<encrypted password></p> |
| security.propertyencrypter.class | Valid values = <your_property_encrypter_class> Default is not set | <p>This class will be used for encrypting and decrypting properties specified in the yfs.properties, yifclient.properties, and management.properties files. Use this property to encrypt sensitive data, such as user IDs and passwords. Properties starting with "encrypted:" are automatically decrypted at run-time.</p> <p>For more information about encryption through property files, refer to the <i>Selling and Fulfillment Foundation: Extending Transactions Guide</i>.</p> <p>Example: yfs.dblogin.datasource.name=encrypted:<encrypted value></p> |
| Service Definition Framework (SDF) | | |
| yfs.smtp.session.reaptime yfs.jms.session.reaptime | Default = 10*60 seconds | This is the frequency at which the JMS/SMTP connection reaper thread will examine the connection pool for connections that are eligible to be closed. The connection will be eligible to be closed if the connection is idle for the number of seconds specified for the reaptime. This property is specified in seconds. |
| yfs.smtp.connectionpool.enable | Valid values = true or false Default = true | <p>To disable SMTP connection pooling, set this property to false. Sterling Commerce recommends setting this property to true for performance reasons.</p> <p>Example: yfs.smtp.connectionpool.enable=true</p> |

| Property | Values | Description |
|---------------------------------|--|--|
| yfs.emailer.class | | The class that handles automated e-mail communication to and from this application. Example: yfs.emailer.class=com.yantra.util.YFCEmailerImp |
| yfs.email.template.encoding | <encoding type> Default = UTF-8 | Handles encoding for e-mail communication to and from this application. Example: yfs.email.template.encoding=UTF-8 |
| yfs.eof.wait.time | <number of seconds> Default = 0 | Defines how many seconds after receiving the EOF message the Service Definition Framework should wait before executing this service. Example: yfs.eof.wait.time=30 |
| yfs.xml.uriresolver | <class name> | This class can be used to provide a custom URIResolver during XSL processing. Refer to the Java documentation for more information on this interface. If this property is not present, then the default implementation provided by the XSL processor is used. If set to com.yantra.interop.util.YantraDefaultURIResolver an attempt is made to resolve the URI within the classpath. If set to another class name, this class is used in place of the YantraDefaultURIResolver. The class given here must implement the javax.xml.transform.URIResolver interface. Example: yfs.xml.uriresolver=com.yantra.interop.util.YantraDefaultURIResolver |
| yfs.sync.jms.request.timetolive | <number of seconds> Default = responseTimeout value of the receiver | Request queue timetolive in seconds for Synchronous JMS messages. Use this property to set the timetolive value for Synchronous JMS Messages. Note that, this property is a global level override and is applicable for all the services using Synchronous JMS. WARNING: If this property value is set beyond the responseTimeout value, it may result in unprocessed request messages in the reply/response queue. By default, in order to clean up the un-processed messages from the reply/response queue, the timeToLive for request messages is set to the responseTimeout value of the receiver. Example: yfs.sync.jms.request.timetolive= |

| Property | Values | Description |
|---|--|---|
| yfs.sync.jms.request.timeToLive.<SERVICE_NAME> | <number of seconds> Default = responseTimeout value of the receiver | Request queue timeToLive in seconds for Synchronous JMS messages. Use this property to set the timeToLive value for Synchronous JMS Messages. Note that, this property is applicable only for the specified service. WARNING: If this property value is set beyond the responseTimeout value, it may result in un-processed request messages in the reply/response queue. By default, in order to clean up the un-processed messages from the reply/response queue, the timeToLive for request messages is set to the responseTimeout value of the receiver. Example: yfs.sync.jms.request.timeToLive.<SERVICE_NAME>= |
| yfs.reprocess.MaxErrorMessageLength | Valid value = integer Default = 4000 | Setting this property will decide the length to which the error message to be stored in YFS_REPROCESS_ERROR table should be truncated. Example: yfs.reprocess.MaxErrorMessageLength= |
| yfs.eof.wait.time | Valid values =<number of seconds> Default = 300 | Set this property to indicate the time interval (in seconds) an EOF message must wait to lapse after receiving an EOF message before executing the service. Example: yfs.eof.wait.time=600 |
| Rich Client Platform (RCP) | | |
| yfs.rcp.devmode | Valid values = Y or N Default = N | Set the property to 'Y' if you want to run the Rich Client Platform application in RCP dev mode. When a Rich Client Platform application is run in the RCP dev mode, the server-side RCP validations such as commands validation for security purposes are not done. Example: yfs.rcp.devmode=Y |
| Sterling Sensitive Data Capture Server (SSDCS) | | |

| Property | Values | Description |
|---|--|--|
| yfs.ssdcs.url | Valid value = https://<host>:<port>/ssdcs Default = https://<host>:<port>/ssdcs | Contains the URL to access the SSDCS application. This URL is not provided out of the box. You must enter the SSDCS URL in this file to enable Selling and Fulfillment Foundation to contact SSDCS. The variables <host> and <port> are placeholders that you must replace with the name of the machine (not the IP address) and with the port. Do not change anything else in the URL. Example: yfs.ssdcs.url=https://sterlingcommerce.com:8080/ssdcs |
| yfs.ssdcs.servlet | Valid value = /tokenize | Contains the default servlet path that can be used in UI coding. It is recommended that you do not change this value. |
| yfs.ssdcs.jsp | Valid value = /jsp/ssdcs_tokenize_pan.jsp | Contains the default jsp path that can be used in UI coding. It is recommended that you do not change this value. |
| yfs.ssdcs.tokenize.cc | Valid values = Y or N Default = Y | A Y/N property to determine if credit cards have to be tokenized. Set the property to N if you do not want to make tokenization calls for credit cards. Example: yfs.ssdcs.tokenize.cc=Y |
| yfs.ssdcs.tokenize.svc | Valid values = Y or N Default = Y | A Y/N property to determine if stored value cards have to be tokenized. Set the property to N if you do not want to make tokenization calls for stored value cards. Example: yfs.ssdcs.tokenize.svc=Y |
| sc.access.token.expire.in.seconds | Valid value = <number of seconds> Default = 600 seconds | This property indicates the default expire time (in seconds) for an access token. Example: sc.access.token.expire.in.seconds=600 |
| sc.access.token.max.allowed.expire.in.seconds | Valid value = <number of seconds> Default = 1800 seconds | This property indicates the maximum allowed expire time (in seconds) for an access token. It can be used to set the upper limit on allowed expire time for an access token. Example: sc.access.token.max.allowed.expire.in.seconds=1800 |

| Property | Values | Description |
|--|--|---|
| System Management | | |
| rmi.portrange | <minport> - <maxport> Default = 0 | In a deployment with servers in two different network zones, the firewall between them must be configured to allow Remote Method Invocation (RMI) communication between them. When you specify a port range, a single, unused port in that range will be selected for use. You can specify multiple ranges separated by a comma. If an open port cannot be found, an exception will occur. If you specify 0, any available port in the RMI range can be selected for use. Do not specify the 0-1023 range or the 1024-49151 range. Example: rmi.portrange=49152-49162 |
| Note: The next four properties are for the Health Monitor. The health monitor will raise the configured alerts when the api/appServer response time or the agent Pending Job count crosses the threshold. | | |
| yantra.hm.api.threshold | <number of milliseconds> Default = 20000 milliseconds | api threshold in milliseconds, this is the system default value used for average response time for appservers. If this value is not specified 20000 milliseconds will be assumed. |
| yantra.hm.agent.threshold | <job size> Default = 10000 | System default value used for the pending jobs size for agents. If this value is not specified, 10000 will be assumed. |
| yantra.hm.appserver.threshold | <number of milliseconds> Default = 8000 milliseconds | Appserver threshold in milliseconds, this is the system default value used for average response time for api's/services. If this value is not specified 8000 milliseconds will be assumed. |
| yantra.hm.purge.interval | <number of days> Default = 30 days | Health Monitor purge interval, this is the system default value used for purging heartbeat and snapshot records. if this value is not specified 30 days purge interval is assumed. |
| yantra.statistics.collect | Valid values = n, N, y, or Y Default = N | Property to turn on Statistics logging. Example: yantra.statistics.collect=Y |

| Property | Values | Description |
|------------------------------------|--|---|
| yantra.statistics.persist.interval | Valid values for minutes (M/m) = 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30 Valid values for minutes (H/h) = 1, 2, 3, 4, 6, 8, or 12 | Property to determine statistics logging time interval. Format of the property is Xm or Xh where X is an integer between 1 and 60 and M/m for minutes or H/h for hours. If any unrecognized value or unit is specified, it will default to 10m (minutes). If a value of 61m or greater is specified, it will be reduced to 60m. If a value of 25h or greater is specified, it will be reduced to 24h. If the units are minutes (M/m), then the value is rounded up or down to the nearest equal divisor of 60 minutes. If the units are hours (H/h), then the value is rounded up or down to the nearest equal divisor of 24 hours. Example: yantra.statistics.persist.interval=10m |
| systemlogger.rotatelogs | | Flag indicating whether to rotate the system log after it has reached its maximum size. If this flag is set to false the below two attributes will be ignored and logger will continue to write in just one log file. Set this flag to true to let the below properties take effect. Example: true |
| systemlogger.maxlogsize | | Maximum number of write operations after which the log messages will be logged in a new log file. Example: 100000 |
| systemlogger.maxnumlogs | | Maximum number of log files after which the oldest log file should be deleted. If the value of this flag is set 10, the 11th log file is created and the first log file will be removed. Example: 10 |

| Property | Values | Description |
|--------------------------------|--|---|
| yfs.heartbeat.refresh.interval | Valid values for minutes (M/m) = 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30 Valid values for minutes (H/h) = 1, 2, 3, 4, 6, 8, or 12 Default = 10 m | Property to determine heartbeat refresh time interval. Format of the property is Xm or Xh where X is an integer between 1 and 60 and M/m for minutes or H/h for hours. If any unrecognized value or unit is specified, it will default to 10m (minutes). If a value of 61m or greater is specified, it will be reduced to 60m. If a value of 25h or greater is specified, it will be reduced to 24h. If the units are minutes (M/m), then the value is rounded up or down to the nearest equal divisor of 60 minutes. If the units are hours (H/h), then the value is rounded up or down to the nearest equal divisor of 24 hours. Example: yfs.heartbeat.refresh.interval=10m Note: Sterling Commerce recommends that the value of this property should be set greater than the value of the yantra.statistics.persist.interval property. |
| User Interface | | |
| yfs.cache.size.regionmap | Default =1000 | Set this property to set the cache size, which maintains a LRU cache of RegionMaps. If a RegionMap needs to be inserted into the cache beyond this size, the least recently accessed RegionMap is removed from the map to provide space for the new RegionMap. |

| Property | Values | Description |
|------------------------------|---|---|
| yfs.config.lookuplimit | Valid values =<integer> Default = 75 | <p>Set this property to determine number of records needed before a drop down list is displayed as a lookup text field.</p> <p>Example: yfs.config.lookuplimit=75</p> <p>Note: The behavior of this property is described below:</p> <p>When the value of the yfs.config.lookuplimit is not set:</p> <ul style="list-style-type: none"> ◆ If the number of records is 75, only the lookup text field is displayed. ◆ If the number of records is less than 75 but greater than 20, the drop down list and lookup text field are displayed. ◆ If the number of records is less than or equal to 20, only the drop down list is displayed. <p>When the yfs.config.lookuplimit is set to a value that is less than 20:</p> <ul style="list-style-type: none"> ◆ If the number of records is greater than the set value, only the lookup text field is displayed. ◆ If the number of records is less than or equal to the set value, only the drop down list is displayed. <p>When the yfs.config.lookuplimit is set to a value that is greater than 20 but less than 75:</p> <ul style="list-style-type: none"> ◆ If the number of records is less than the set value but greater than 20, the drop down list and lookup text field are displayed. ◆ If the number of records is greater than the set value, only the lookup text field is displayed. ◆ Number of records is less than 20, only the drop down list is displayed. |
| yfs.disable.webpages.caching | Valid values = Y, y, N, or n Default = N | <p>Setting this property to 'Y' will disable caching of Application pages.</p> <p>Example: yfs.disable.webpages.caching=N</p> |
| yfs.rcp.pca.updates.dir | | <p>Set the property to specify the root folder under which the updates for the PCAs are located. It can be a shared folder on a network.</p> <p>Example: yfs.rcp.pca.updates.dir=<updates_directory></p> |

| Property | Values | Description |
|-------------------------------|--------------------------------------|---|
| yfs.rcp.pca.updates.cache.dir | | <p>Set following property to a local cache directory, where the updates need to be cached.</p> <p>IMPORTANT: The updates cache directory MUST NOT point to same location as the yfs.rcp.pca.updates.dir. The cache directory is a working directory and involves file I/O e.g. file creations and deletions, etc.</p> <p>Example: yfs.rcp.pca.updates.cache.dir=<updates_directory></p> |
| yfc.ui.ListPageSize | <number of records> | <p>Set this to determine number of records shown on paginated list page.</p> <p>Example: yfc.ui.ListPageSize=30</p> |
| yfs.login.redirect.url | | <p>If the system needs to redirect to some custom url upon error in the login page, this property needs to be set to this url. If not specified, it will redirect to the default login page. This property can be specified in yfs.properties_ext file.</p> <p>Example: yfs.login.redirect.url=<REDIRECT_URL></p> |
| yfs.ui.MaxRecords | <number of records> Default = 200 | <p>This property sets the number of records displayed on a list screen. Increase the application server JVM heap settings if these parameters are increased. Change will affect search limits for all users. The value of this property should not exceed 999.</p> <p>Example: yfs.ui.MaxRecords=200</p> |
| yfs.ui.queryTimeout | <number of seconds> Default = 60 | <p>This property sets the number of seconds the Application Console will wait for an SQL Statement to execute. If the limit is exceeded, an SQLException is thrown. Set to zero means unlimited.</p> <p>Example: yfs.ui.queryTimeout=60</p> |
| yfs.ui.defaultEncoding | Default = UTF-8 | <p>This property controls the encoding of the HTML sent to the client.</p> <p>Example: yfs.ui.defaultEncoding=UTF-8</p> |

| Property | Values | Description |
|----------------------------------|---|--|
| yfs.config.java.plugin.codebase | Valid values =<location of the plugin> Default =http://java.sun.com/update/1.4.2/jinstall-1_4_2_03-windows-i586.cab | If you wish to install the Java plugin (used for the Applications Manager and HSDE's) in intranet environments, configure this parameter to point to the location of the plugin. If not passed, it defaults to http://java.sun.com/update/1.4.2/jinstall-1_4_2_03-window s-i586.cab. For more details, refer to http://java.sun.com/products/archive/j2se/1.4.2/jre/installwindows.html. Example: yfs.config.java.plugin.codebase=http://java.sun.com/update/1.4.2/jinstall-1_4_2_03-windows-i586.cab |
| yfs.config.password.noprompt | Valid values = Y or N Default = N | This enables the ability in the Applications Manager to health monitor to popup a message asking the user to log back via the console. Example: yfs.config.password.noprompt=N |
| yfs.rcp.ui.pagesize | Default = 30 | This property sets the maximum number of records to display on a single page in the Table on the Search List screen. This property is used by PCA Applications. Example: yfs.rcp.ui.pagesize=30 |
| yfs.gridLayout.maximumRecords | Default is not set | This property is used to limit the number of records shown in grid screens in Mobile Application. For vt220 clients, because of memory implications, this property shouldn't be set to more than 7. If this property is not set or is not a positive integer, the default value would be taken as 7. Example: yfs.gridLayout.maximumRecords= |
| yfs.wms.console.show.node.lookup | Default = N | This property decides how the Select Node user interface will appear in the Application Consoles. If the value of this property is set to N, a drop-down list is displayed against the Select Node field. If the value of the property is set to Y, a lookup icon is displayed against the Select Node field. Example: yfs.wms.console.show.node.lookup=N |

| Property | Values | Description |
|--|--------------------------------------|---|
| yfs.console.login.error.showDetail=N | | If this property is set to Y, error message details will be shown on the console login screen when the # login fails. The default value is N, which means that the error detail will NOT be shown by default. |
| Warehouse Management | | |
| yfs.allow.reuse.previouswave.slots | Valid values = Y or N Default = Y | When dynamic slotting is configured, for the second and subsequent wave releases, the previously slotted locations are considered in order to minimize the number of locations. If this property is set to N, the previously slotted locations are not considered for the subsequent wave releases. Example: yfs.allow.reuse.previouswave.slots=Y |
| yfs.solver.iterations.wavecreate | | The number of solver iterations for Create Wave. Use this property to set the number of solver iterations through which the create wave Agent will undergo to optimize wave creation. Example: yfs.solver.iterations.wavecreate=1 |
| yfs.cancelwave.offline.taskcount.greaterthan | | Cancellation of wave happens in background depending upon the value of the property yfs.cancelwave.offline.taskcount.greaterthan. If the value of the property is for ex: 500 then cancellation of a wave, which has open tasks more than 500 included in it then the wave gets cancelled in offline mode. Waves that have open tasks equal to or less than 500 gets cancelled online. Example: yfs.cancelwave.offline.taskcount.greaterthan=-1 |
| yfs.cancelwave.offline.shipmentcount.greaterthan | | Cancellation of wave happens in background depending upon the value of the property yfs.cancelwave.offline.shipmentcount.greaterthan. If the value of the property is for ex: 500 then cancellation of a wave, which has shipments more than 500 included in it then the wave gets cancelled in offline mode. Waves that have shipments equal to or less than 500 gets cancelled online. Example: yfs.cancelwave.offline.shipmentcount.greaterthan=-1 |

| Property | Values | Description |
|--|--------|---|
| yfs.cancelmove.offline.taskcount.greaterthan | | <p>Cancellation of move request happens in background depending upon the value of the property</p> <p>yfs.cancelmoverequest.offline.taskcount.greaterthan. If the value of the property is for ex: 500 then cancellation of a move request, which has open tasks more than 500 included in it then the move request gets cancelled in offline mode. Move requests that have open tasks equal to or less than 500 gets cancelled online.</p> <p>Example:</p> <p>yfs.cancelmove.offline.taskcount.greaterthan=-1</p> |
| yfs.canceltask.onshortpick.tasktypelist | | <p>Cancellation of specified tasktypes happens for the remaining quantity on a short pick depending upon the value of the</p> <p>yfs.canceltask.onshortpick.tasktypelist. You can specify multiple tasktype values for this property; however, values must be separated by commas.</p> <p>Example:</p> <p>yfs.canceltask.onshortpick.tasktypelist=<tasktype1>,<tasktype2>,<tasktype3></p> <p>Only tasktypes specified in</p> <p>yfs.canceltask.onshortpick.tasktypelist= are canceled, and if tasktype is not specified in this property, the task is moved to hold status.</p> |
| yfs.containerization.maxshipmentsinoneround | | <p>The containerizeWave API will use this property to select number of shipments to be considered for creating the outbound containers in one round. API will consider default value as 75 if value is not specified for this property.</p> <p>Example:</p> <p>yfs.containerization.maxshipmentsinoneround=75</p> |
| yfs.install.createWave.delayhours | | <p>The CREATE WAVE FOR SHIPMENT GROUP agent uses this property to increase the Available date (in hours) of YFS_Task_Q records with transaction key</p> <p>"CREATE_WAVE_TASK_Q.4001" against which no waves could be created.</p> <p>Agent considers default value as 1 if value is not specified for this property. If set to -1, the YFS_Task_Q records with transaction key</p> <p>"CREATE_WAVE_TASK_Q.4001" for which no waves are generated, will be deleted.</p> <p>Example:</p> <p>yfs.install.createWave.delayhours=1</p> |

| Property | Values | Description |
|--|-----------------------|--|
| yfs.createwave.dbRead.numShipmentLines | Default = 20000 | The CREATE WAVE agent uses this property to read maximum number of shipment lines from database in a single query. Example: yfs.createwave.dbRead.numShipmentLines=20000 |
| yfs.createwave.process.numShipmentLines | Default = 50000 | The CREATE WAVE agent uses this property in order to determine the maximum number of shipment lines for optimal memory utilization during wave creation. Example: yfs.createwave.process.numShipmentLines=50000 |
| yfs.releasetask.ignore.wavesequences | Valid values = Y or N | If this property is set to "Y", the RELEASE TASK agent ignores the wave sequence when releasing tasks. For example, if there are two waves having the same priority, but different sequence numbers, and if the tasks for the wave having a lower sequence number cannot be released, the tasks for the wave having a higher sequence number will be released. Example: yfs.releasetask.ignore.wavesequences=Y |
| yfs.wms.resourceplanning.QueueName yfs.wms.resourceplanning.ProviderURL yfs.wms.resourceplanning.QCFLookUp | | Set these JMS properties to enable resource planning move. Examples: yfs.wms.resourceplanning.QueueName=DefaultAgentQueue yfs.wms.resourceplanning.ProviderURL=t3://localhost:7001 yfs.wms.resourceplanning.QCFLookUp=AGENT_QCF |
| yfs.wms.zonetransattr.refreshrate | Default = 600 seconds | getZoneDetails API uses the below mentioned property to refresh the zone attributes: available volume, available weight, pending volume and pending weight in the table YFS_TRAN_ZONE_ATTRS. These zone attributes are aggregates of the location attributes. Zone attributes # are refreshed after specified time intervals which is defined by this property. Example: yfs.wms.zonetransattr.refreshrate=600 |

| Property | Values | Description |
|--|--|---|
| yfs.releaseWave.SortShipmentsByDateAndKey | Valid values = Y or N Default = N | During Release wave, if shipments need to be allocated quantity based on requested shipment dates, set this property to "Y". Example: yfs.releaseWave.SortShipmentsByDateAndKey=N |
| yfs.default.inventorystatus.for.newinventory | | In case of counting through mobile application if user counts new inventory, system currently displays first inventory status of the node as defaulted. If this behavior needs to be overridden, then the below property needs to be set with valid inventory status which will be used as default inventory status for new inventory. If this property is not set, system will retain the current behavior. Example: yfs.default.inventorystatus.for.newinventory= |
| yfs.closemanifest.online | Valid values = Y or N | Manifesting options. Set this property to N to close manifest asynchronously using the CLOSE_MANIFEST Agent. Example: yfs.closemanifest.online=Y |
| yfs.confirmbatch.online | Valid values = Y or N | Batch completion options. Set this property to N to complete batch asynchronously using the REQ_BATCH_COMPLETION Agent. Example: yfs.confirmbatch.online=Y |
| yfs.reopentask.time | Default = 1 hour | The re-open task agent will change the status of a suggested task back to open status, if the task is not modified for the time specified by this property. This property is specified in hours. Example: yfs.reopentask.time=1 |
| yfs.serial.receiving | Valid values = Y or N | Set this property to 'Y' to process one serial at a time during receiving i.e on scanning one serial user is taken to Disposition Code entry screen. Example: yfs.serial.receiving=N |
| yfs.releasetaskagent.mode | Valid values = 01: Location Driven. All the pending tasks from the location are evaluated. 02: Reference (eg. ShipmentNo) Driven. All the pending tasks for the reference are evaluated. Default = 01 | releaseTaskAgent uses this property to select the mode of execution for releasing tasks. Example: yfs.releasetaskagent.mode=01 |

| Property | Values | Description |
|--|--------------------------------------|---|
| yfs.retain.pack.tasks.for.minute | Default = 60 minutes | <p>In case of item driven packing, shipment level transaction locks are created in order to reserve the shipment against the packer. Once the shipment packing is complete, these transaction locks are deleted from DB.</p> <p>If for some reason packing for this shipment cannot be completed by the packer, these transaction locks block other users from packing this shipment.</p> <p>This property can used to define maximum time, in minutes, for which the shipment level transaction locks are retained. If it is set to 120, and a user scans a Pallet in Pack HSDE, transaction locks created 120 minutes before will be deleted so that the unpacked blocked shipment becomes available to other users.</p> <p>Example: yfs.retain.pack.tasks.for.minutes=60</p> |
| yfs.override.user.constraints.for.manually.assigned.task | Valid values = Y or N Default = N | <p>Based on this property, the maximum user constraints defined for a task type will be evaluated. If task is manually assigned to a user and this property is set to Y, the maximum user constraint defined for task type will not be honoured. To retain the existing behavior, set this property to N.</p> <p>Example: yfs.override.user.constraints.for.manually.assigned.task=N</p> |
| yfs.displayTaskInfoDetails | Valid values = Y or N Default = N | <p>Set this property to 'Y' to show the task details (Pick Location, Item ID and Quantity) in RF mobile terminal when tasktype is defined to say pick onto equipment. This provides user an ability to determine whether he needs to carry a pallet/case to pick inventory even before he goes to the pick location.</p> <p>Example: yfs.displayTaskInfoDetails=N</p> |
| yfs.override.pendingallocations.for.adhocmoves | Valid values = Y or N Default = N | <p>If Delayed Inventory Allocation Support is configured in a Node, the default behavior is that system will wait to complete the pending allocation before creating new tasks. Setting this property to Y allows ad-hoc move operations on Mobile Application to be processed by the system without waiting for the Allocate Task agent to complete the pending allocation.</p> <p>Example: yfs.override.pendingallocations.for.adhocmoves=Y</p> |

| Property | Values | Description |
|--|--------------------------------------|---|
| yfs.allow.container.quantity.adjustment | Valid values = Y or N | In case of picking through mobile application if user picks a full LPN, system currently doesn't provide an option to adjust the discrepancies. To allow user to adjust the discrepancies in the license plates, the property needs to be set to Y. If this property is not set or set to N, system will retain the current behavior. Example: yfs.allow.container.quantity.adjustment= |
| yfs.determine.shipmentgroup.for.shipment | Valid values = Y or N Default = Y | Set this property to "N" to prevent the shipment group determination during shipment creation. The shipment group for these shipments is determined later by the The CREATE WAVE FOR SHIPMENT GROUP agent. |
| yfs.split.reference.across.totes | Valid values = Y or N | If this property is set to "Y", the system throws a warning when a user picks the same reference, such as shipment, move request, or batch into multiple totes or target LPNs when performing cart picking through the Sterling Supply Chain Mobile Application. |

dbclassCache.properties.*

The dbclassCache.properties file contains a list of the Selling and Fulfillment Foundation tables and the associated DBCache class. Every table listed will be registered with the cache manager with the default cache sizes. The dbclassCache.properties file also contains the global settings for the DBCache classes, and additional properties that can be uncommented and set as needed. Do not make changes directly to the dbclassCache.properties file; instead, use the customer_overrides.properties file.

| Property | Values | Description |
|-----------------------------|---|--|
| sci.globalcache.select.size | Any positive integer, defaults to 10000 | Default maximum size of the "select" database cache for a given table. This is the number of select statements that return a single record that will be cached. |
| sci.globalcache.list.size | Any positive integer, defaults to 10000 | Default maximum size of the "list" database cache for a given table. This is the number of select statements that return a multiple records that will be cached. |
| sci.globalcache.count.size | Any positive integer, defaults to 10000 | Default maximum size of the "count" database cache for a given table. This is the number of select statements that count records that will be cached. |

| Property | Values | Description |
|-----------------------------|---|--|
| sci.globalcache.object.size | Any positive integer, defaults to 10000 | Default maximum size of the “object” database cache for a given table. This is the total number of resulting database objects across the “select” and “list” caches that will be cached. If this limit is reached, all database caches for the table are dropped |

The `dbclassCache.properties` file also lists the cacheable tables, in the form of `<TableName>.class=<cache implementation>`. This would only be changed/added by applications.

Each table also supports overriding the defaults provided in the `globalcache` settings. For example, using `YFS_COMMON_CODE` as the table name:

| Property | Values | Description |
|-----------------------------|---|---|
| YFS_COMMON_CODE.enabled | true or false. Default value is true. | Whether the database cache is enabled. Set to false to disable caching. |
| YFS_COMMON_CODE.select.size | Any positive integer, defaults to value of <code>sci.globalcache.select.size</code> | Override the “select” cache size for the given table. |
| YFS_COMMON_CODE.list.size | Any positive integer, defaults to value of <code>sci.globalcache.list.size</code> | Override the “list” cache size for the given table. |
| YFS_COMMON_CODE.count.size | Any positive integer, defaults to value of <code>sci.globalcache.count.size</code> | Override the “count” cache size for the given table. |
| YFS_COMMON_CODE.object.size | Any positive integer, defaults to value of <code>sci.globalcache.object.size</code> | Override the “object” cache size for the given table. |

Sandbox.cfg

`Sandbox.cfg` contains properties related to system and database information.

Note: `Sandbox.cfg` contains ports used by multiple Sterling applications in addition to Selling and Fulfillment Foundation. For Selling and Fulfillment Foundation, only the ports defined in the following table are in use by the application. Other ports defined in `sandbox.cfg` are not in use and do not conflict with ports that may be required by other applications running on the same host. Specifically, Selling and Fulfillment Foundation uses only a subset of the database ports: `DB_PORT`, `DB2_PORT`, `MSSQL_PORT`, `ORA_PORT`, based on your database vendor. Your application server vendor requires additional ports, not described here. Refer to the vendor’s documentation to determine the ports that may be required by the application server. The following table provides descriptions of the ports used by Selling and Fulfillment Foundation. The table does not provide information about ports used by other Sterling applications.

Note: The `sandbox.cfg` file is not used at runtime by the product. If you change a parameter in the `sandbox.cfg` file at any time, you must run the `setupfiles` script so that the runtime property files are re-created with the updated values.

| Property | Description |
|-----------------------------------|--|
| Installation | |
| ADDITIONAL_ANT_JAVA_TASK_ARGS | During the installation, you can pass JVM-specific arguments to avoid out-of-memory errors. Example: <code>-Xms1024m -Xmx1408m -XX:MaxPermSize=512m</code> |
| ADDITIONAL_ANT_COMPILER_TASK_ARGS | During installation, you can pass initial and maximum JVM-specific arguments to avoid out-of-memory errors. Example: <code>ADDITIONAL_ANT_COMPILER_TASK_ARGS=-J-Xms1024m -J-Xmx1536m</code> |
| JVM_LOC | Source of downloaded JDK files, external to Selling and Fulfillment Foundation application files. Example: <code><jdk_dir></code> |
| JAVA_HOME | Points to the location of the Java SDK that is used for the installation. This location can be the directory with the downloaded JDK files (<code>JVM_LOC</code>) or it can be the Selling and Fulfillment Foundation directory to which they have been copied during installation. Example: <code><jdk_dir></code> or <code><INSTALL_DIR>/jdk</code> |
| DB_CLEAN | Deletes all objects from the database for the user. The GUI installer sets this property to false. Valid values = true or false Default: true |
| ENTITY_GEN_LOGLEVEL | Specify VERBOSE to enable logging of verbose messages during entity class generation. Valid values = VERBOSE or INFO Default: INFO |
| Agent | |
| AGENT_JAVA_HOME | Overrides the java that the agents use. For instance, if, with WebSphere, you wanted to use the IBM jdk, you could set this parameter to a new value. |

| Property | Description |
|---------------------------|--|
| Internal | |
| COPY_FCXML_TO_REPOSITORY | <p>Only used at install time.</p> <p>Required. Installs the files related to the database factory defaults in the <INSTALL_DIR>/repository/factorysetup directory. This enables you to delete the <INSTALL_DIR>/installed_data/<package_name>/factorysetup directory after installation, as it contains other files that take up space.</p> <p>The GUI installer sets this property to true.</p> <p>Valid values = true or false</p> <p>Default: true</p> <p>Note: This property should be set to true.</p> |
| DB_DATA | <p>Only used at install time. Database name to connect to.</p> <p>Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table).</p> |
| DB_DRIVERS | <p>Only used at install time. Full path to JDBC driver file.</p> <p>Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table).</p> |
| DB_HOST | <p>Only used at install time. Database host to connect to.</p> <p>Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table).</p> |
| DB_JAR_DIR | <p>Only used at install time.</p> <p>Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table).</p> |
| DB_PASS | Database password to connect with. |
| DB_POOL | Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table). |
| DB_PORT | <p>Only used at install time. Database listener port.</p> <p>Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table).</p> |
| DB_USER | <p>Database login ID to connect with.</p> <p>Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table).</p> |
| DB_VENDOR | <p>Required. The DB vendor to use. Valid values are Oracle, DB2, or MSSQL.</p> <p>Note: To change this database property after installation, use the database specific properties (see the properties in the Database section of this table).</p> |
| INSTALL_DIR | Required. Directory in which to install. |
| JDBC_DRIVER | Path to database driver file. |
| PLATFORM_AFC_BUILD_NUMBER | <p>It is recommended that you do not change this value.</p> <p>Example: 5000</p> |

| Property | Description |
|--|---|
| PLATFORM_AFC_LIC_PROD_VERSION | It is recommended that you do not change this value. Example: 5_0 |
| PLATFORM_AFC_PRODUCT_LABEL | It is recommended that you do not change this value. Example: platform_afc |
| SI_LICENSE_AVAILABLE | Indicator of if a license is being passed in and is required for installation. Valid values are Yes or No (default). |
| NOAPP_HOME CLASS_DIR HOME_DIR VENDORS_DIR DEPLOYED_APP_DIR BIN_DIR APPBEANS_DIR SVC_DIR DIST_DIR | These properties should only be changed as a group. |
| SQLProxy Tool | To set up the SQL Proxy tool, add the following properties to <code>customer_overrides.properties</code> prior to starting the application server or noapp server. These will override settings for these properties that are in <code>jdbc.properties</code> . For more information, see <i>Selling and Fulfillment Foundation: Extending the Database Guide</i> . |
| <code>jdbcService.proxyLoggingEnabled</code> | Valid values are Y or N. Specifies whether SQLProxy logging is enabled. Example: <code>jdbcService.proxyLoggingEnabled=Y</code> |
| <code>jdbcService.proxySourceLogging</code> | Valid values are Y or N. Used when enabling SQLProxy. Example: <code>jdbcService.proxySourceLogging=Y</code> |
| <code>jdbcService.proxyLogDir</code> | Valid values are Y or N. Specifies the directory where the trace files will be generated. Example: <code>jdbcService.proxyLogDir=<INSTALL_DIR>/logs/sqlproxylogs</code> |

| Property | Description |
|-----------------------------|--|
| Database | <p>These are database-related properties that can be changed after installation. Use the vendor-specific properties in this section to make changes to your installed database's properties.</p> <p>Note: You cannot change the database provider for a specific smcfs instance after installation.</p> |
| DB_SCHEMA_OWNER | Default schema/schema-owner for the provided login ID. |
| JDBC_VENDOR | JDBC driver vendor. Used when multiple vendor are available. Default is Microsoft. |
| DB2_DATA | If using DB2, the database name to connect with. |
| DB2_HOST | If using DB2, the database host to connect to. |
| DB2_PASS | If using DB2, the database password to connect with. |
| DB2_PORT | If using DB2, the database listener port. |
| DB2_USER | If using DB2, the database login ID to connect with. |
| MSSQL_DATA | If using MSSQL, the database name to connect with. Valid for both MSSQL2000 and MSSQL2005. |
| MSSQL_HOST | If using MSSQL, the database host to connect to. Valid for both MSSQL2000 and MSSQL2005. |
| MSSQL_PASS | If using MSSQL, the database password to connect with. Valid for both MSSQL2000 and MSSQL2005. |
| MSSQL_PORT | If using MSSQL, the database listener port. Valid for both MSSQL2000 and MSSQL2005. |
| MSSQL_USER | If using MSSQL, the database login ID to connect with. Valid for both MSSQL2000 and MSSQL2005. |
| ORA_HOST | If using Oracle, the database host to connect to. |
| ORACLE-NLS_LENGTH_SEMANTICS | <p>Indicates the type of length semantic to be used for Oracle database, when using the dbverify tool.</p> <p>Valid values = CHAR or BYTE</p> <p>Default: BYTE</p> <p>Example: BYTE</p> <p>Note: If the database or the specific session in which database was created has length semantic as CHAR, this property should be set to CHAR before running the dbverify tool.</p> |
| ORA_PASS | If using Oracle, the database password to connect with. |
| ORA_PORT | If using Oracle, the database listener port. |
| ORA_TS | <p>Indicates the kind of text search index to be used for Oracle database, when using the dbverify tool.</p> <p>Valid values = CTXCAT or CONTEXT</p> <p>Default: CTXCAT</p> <p>Example: CTXCAT</p> |

| Property | Description |
|--|--|
| ORA_USER | If using Oracle, the database login ID to connect with. |
| DB Pooling | |
| MIN_TRANS_POOL | mssqlPool.initsize |
| MAX_TRANS_POOL | mssqlPool.maxsize |
| Product Documentation Libraries | |
| ACTIVE_DOC_URL | <p>This value determines whether the Help > View Product Documentation menu item invokes the Online Documentation Library (ONLINE_DOC_URL) or the Local Documentation Library (LOCAL_DOC_URL).</p> <p>The Online Documentation Library consists of a Sterling web-hosted Selling and Fulfillment Foundation documentation set in HTML and PDF format.</p> <p>The Local Documentation Library consists of a locally hosted Selling and Fulfillment Foundation documentation set in HTML format.</p> <p>This value is chosen during the GUI and text-based installations, which write the value out to the ONLINE_DOC_URL and LOCAL_DOC_URL properties. In the silent installation, you must define these values, as explained in the <i>Selling and Fulfillment Foundation: Installation Guide</i>. These values can be changed after installation. If the correct URL is not entered, users will receive a “404 Not Found” error when they click the View Product Documentation option on the Help menu.</p> <p>Note: These properties are different from the properties described in this guide for Context-Sensitive Help. Also, the <i>Selling and Fulfillment Foundation: Installation Guide</i> describes connection settings for Context-Sensitive Help.</p> <p>Examples:</p> <p>ACTIVE_DOC_URL = ONLINE (default)</p> <p>ACTIVE_DOC_URL = LOCAL</p> <p>For more information, see the <i>Selling and Fulfillment Foundation: Installation Guide</i>.</p> |
| ONLINE_DOC_URL | <p>The value of this property is the URL for the Sterling-hosted web page that contains the Selling and Fulfillment Foundation Online Documentation Library HTML and PDF format.</p> <p>Example:</p> <p>ONLINE_DOC_URL=http://www.sterlingcommerce.com/Documentation/MCSF90/HomePage.htm</p> <p>This URL cannot be changed. The first time a user visits this Sterling-hosted web page, registration is required. Subsequent visits require only login.</p> |
| LOCAL_DOC_URL | <p>The value of this property is the URL for the local instance of the Selling and Fulfillment Foundation Local Documentation Library in HTML format.</p> <p>Example:</p> <p>LOCAL_DOC_URL=/smcfsdocs/yfsccommon/online_help/en_US/wwhelp/wwhimpl/js/html/wwhelp.htm</p> |

| Property | Description |
|-----------------------|---|
| Implementation | These are properties that can be changed after installation. |
| LOAD_FACTORY_SETUP | Indicates whether or not you want to load factory setup defaults during installation or manually after installation. |
| AUDIT_LOAD_DEFAULTS | Valid values are true or false. When set to true the audits are generated when the loadDefaults script is run. |
| LOG_DIR | Use to override the logging directory. For example, if you want to deploy the ear on another server you could set the parameter to a new value, run setupfiles, then build the ear. The application would then log to the directory you mentioned. |
| MAX_MEMORY | The maximum amount of memory. Example: 512 |
| NO_DBVERIFY | Valid values are true or false. When set to true during installation and install service, dbverify will not be run. This means that Selling and Fulfillment Foundation will not generate DDL to make the database like the XML entity repository. |
| REINIT_DB | Whether Selling and Fulfillment Foundation should initialize the database or not. Valid values are Yes (default) or No. |
| Localization | |
| SUPPORT_MULTIBYTE | Valid values are Y or N. If you are installing on a DB2 or Microsoft SQL Server [®] and need to localize your database using a multi-byte character set, set this flag to Y after installation. This ensures that the database column sizes are large enough to handle the multibyte characters correctly. |
| Multischema | |
| multischema.enabled | Valid values are true or false. During installation, this property in the silent installation file indicates whether this is a multischema installation. If true, the installation looks for a customer-created multischema.xml file, which specifies database information for the Configuration, Metadata, Transaction, and Statistics schemas. Specify this property in lowercase only. Example: multischema.enabled = true For more information, see the <i>Selling and Fulfillment Foundation: Installation Guide</i> . |
| multischema.version | (Required) During installation, this property in the silent installation file indicates which Selling and Fulfillment Foundation version is being installed. Specify this property in lowercase only. Example: multischema.version = 8.5 (recommended with Version 8.5 installation) |
| multischema.file | During installation, this property in the silent installation file indicates the name of the user-defined XML file that contains multischema database information. Specify this property in lowercase only. Example: multischema.file = multischema.xml For more information, see the <i>Selling and Fulfillment Foundation: Installation Guide</i> . |

| Property | Description |
|-------------------------------------|--|
| STERLING_FOUNDATION_PRODUCT_LABEL | (Required) Specifies the product label. Example: STERLING_FOUNDATION_PRODUCT_LABEL = Selling and Fulfillment Foundation |
| STERLING_FOUNDATION_PRODUCT_VERSION | (Required) Specifies the product version you are installing. Example: STERLING_FOUNDATION_PRODUCT_VERSION = 9.0 |
| Application Platform | |
| ANT_DIR | Contains the ant binaries used in the java deployer and other deployment and build scripts. Example: <i>install_dir/ant</i> |
| ANT_VER | The version of the ant released with an application. It is recommended that you do not change this value. Example: 1_6_5 |
| DOC_DIR | The root directory for the XAPI documentation. This is exported in the tmp.sh command. Example: <i>install_dir</i> |
| JAR_DIR | Directory used by install and install3rdParty to store 3rd party software jar files (referenced by the dynamic class loader and tmp.sh for the java classpath). Example: <i>install_dir/jar</i> |
| JAVADOC_COPYRIGHT_INFO_LABEL | It is recommended that you do not change this value. Example: "1999-2010 Sterling Commerce, Inc. All Rights Reserved." |
| JAVADOC_PRODUCT_LABEL | It is recommended that you do not change this value. Example: Platform_Javadocs. |
| JDK64BIT | Specifies whether you're using a 32-bit JDK or a 64-bit JDK. This setting is important for interactive password libraries and other operating system tie-ins. These are non-Java libraries. Default: true (indicates 64-bit JDK) Example: false (indicates 32-bit JDK) |
| LIC_PROD_VERSION | Product version (not build version). It is recommended that you do not change this value. Example: 2.0 |
| ORA_TS_CONTEXT | Indicates the kind of text search index to be used for Oracle database, when using the dbverify tool. Valid values = CTXCAT or CONTEXT Default: CTXCAT Example: CTXCAT |
| PROP_DIR | Path to the properties subdirectory for the an application installation. Example: <i>install_dir/properties</i> |

| Property | Description |
|-----------------|---|
| SYSTEMP_DIR | Derived from the INSTALL_DIR property, which is the user-specified root of the directory structure for an application on the file system. This is the location where temporary files are used. Example: <i>install_dir/tmp</i> |
| XALAN_VER | Used to specify which version of the Xalan jars is being used. When there are multiple JDKs, different versions of these jars are required. It is recommended that you do not change this value. Example: 2_5_2 |
| XERCES_VER | Used to specify which version of the Xerces jars is being used. When there are multiple JDKs, different versions of these jars are required. It is recommended that you do not change this value. Example: 2_6_0 |

A

Applications Manager, 12

C

cache for catalog search index
configuring, 5

L

LDAP user authentication, 11
assumption, 11
for application consoles, 11
modifying weblogic startup file, 12
modifying yfs.properties
customer_overrides.properties, 12
password expiration, 11
change password link, 11
expire in days, 11
properties, 11
log files
setting up, 13
logging
masking sensitive information using log4j utility, 14
properties, 13
log4j configuration file, 13

M

masking sensitive information during logging, 14
memory usage for catalog search index
limiting, 5

O

Oracle database
production environment
setting connection properties, 7

P

password expiration, 11

S

Setting Oracle Database Connection Properties, 7
setting up Selling and Fulfillment Foundation, 5