

Cache Flags

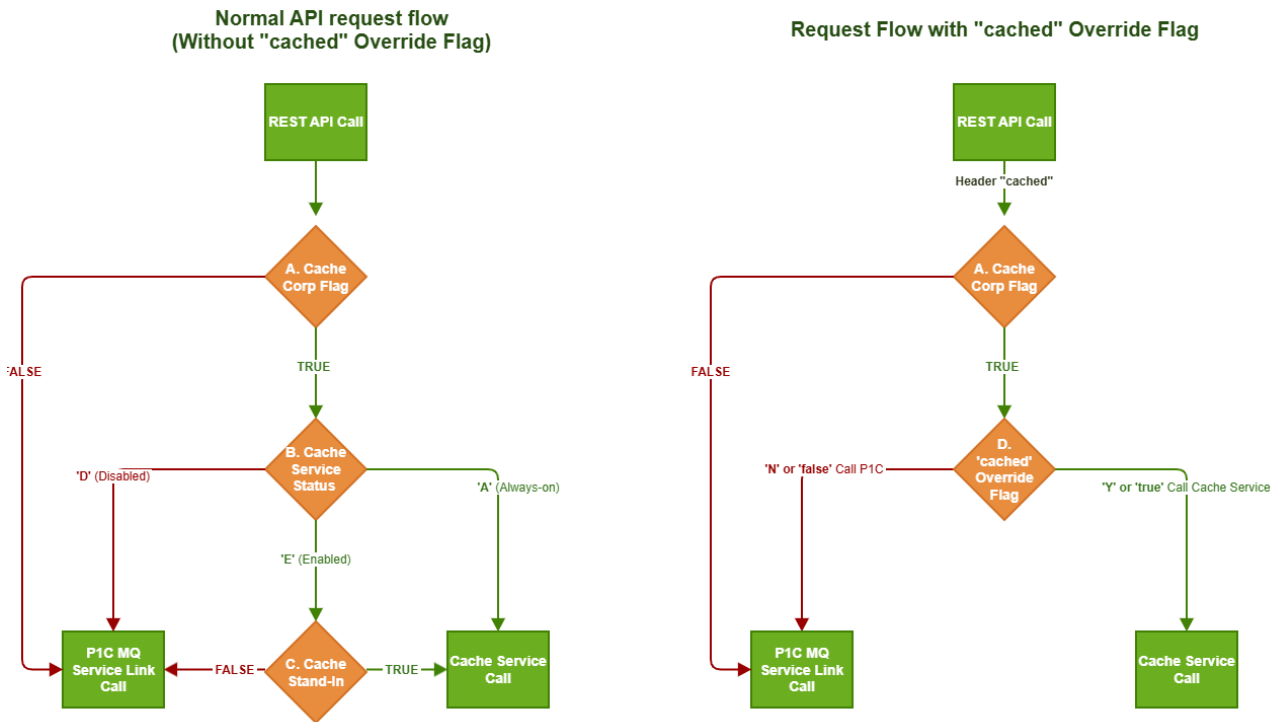
FIS Australasia P1C IC Implementation

Exported on 07/26/2024

Table of Contents

1	Request Flow for Infinity Connect HA APIs.....	3
2	A. Cache Corporation Flag	5
2.1	Configuration.....	5
2.2	Use	5
3	B. Cache Service Status.....	6
3.1	Configuration.....	6
3.2	Use	7
4	C. Cache Stand-In Flag	8
4.1	QCMD Batch Phase	10
4.2	XPH Extended System Status	11
5	Admin Services	14
5.1	Corporation Flag Inquiry (ConfigCache v1.0.0).....	14
5.2	Cache Service Status Flag (ConfigCacheSvc v1.0.0)	16

1 Request Flow for Infinity Connect HA APIs



Infinity Connect APIs can execute P1C Service Links via MQ or Cache services when high availability is enabled. The Infinity Connect application was enhanced to support three flags that are used to decide if the request will be serviced by a P1C Service Link (via MQ) or by a Cache service.

A. The first flag is called 'Cache Corporation Flag' defined at the corporation or at the application level (when multiple corps are supported) and is used to completely switch on/off the Cache module (all Cache services).

B. The second is an indicator called 'Cache Service Status' which is defined at the API level and is used to enable or disable a specific Cache service.

C. The last flag is the 'Cache Stand-In flag' which is only evaluated when the Cache Service Status is set to 'E' (Enabled). This flag will be set to TRUE when NRT batch commands (QCMD) or XPH Service extended processing system status values indicate that the Cache application is to be used during unplanned or planned outages (extended processing phases, release implementations and planned IPLs). For more information about stand-in mode refer to the latest BD section '2.3.3.2.12 Cache Stand-in/Send SAF Requests – based on P1C System Status'.

The Cache service will only be called when the following conditions are met. Logic implemented in the BASEGlobalBacksideAdapter.isCachedCall method:

1. The Cache Corporation flag is enabled (TRUE) AND
2. The Cache Service Status is 'A' (Always-on) OR

3. The Cache Service Status is 'E' (Enabled) AND Cache stand-in mode is enabled (TRUE).

Table below is from BD (v1-03) and shows the only scenarios where a Cache service can be called:

Corporation Cache Flag	Cached Request Override Flag (Client provided)	Service Cache Replica Indicator	Service Cache Replica Stand-in Flag	Cache Application data is used	CACHED-RESPONSE Element
TRUE	Y	A (Always ON)	Any value**	Yes	TRUE
TRUE	Y	E (Enabled)	Any value**	Yes	TRUE
TRUE	Not provided	A (Always ON)	Any value**	Yes	TRUE
TRUE	Not Provided	E (Enabled)	TRUE	Yes	TRUE

** Note: 'Any value' indicates TRUE or FALSE.

Notes:

- The logic above will be implemented in the BASEGlobalBacksideAdapter.isCachedCall method.
- A client consuming IC HA APIs can optionally send a 'cached' HTTP request header with a value of 'Y' to force a Cache service to execute or 'N' to call P1C. This will override other flags (#B and #C) only if the Cache Corporation flag (#A) is enabled. If not provided, then the normal API request flow will run.

2 A. Cache Corporation Flag

Main flag used to enable/disable calls to the cache module. Clients using HA APIs must have the CACHE_ENABLED flag turned on in the SQL scripts configuration.

- Defined in the configuration database.
- Configuration category: BASEGLOBAL.CACHE.CONNECTOR
- XML configuration: <cache-connector-cfg><enabled>true</enabled></cache-connector-cfg>

2.1 Configuration

Located in 'XpressNgServices\service-groups\B2K\Configuration\assembly\database\oracle\p1cic-baseline\data\custom_packs\v2xml\common\'. Sample configurations:

- To configure the flag for all MPMS environments use '\mpmsna\cfg_var_corp.sql'
- For only MPMS SIT environment update '\mpmsna\mpcsit\01.cnctsvcs_cfg_data_load_mpcsit.sql'
- If the variable is defined in both files then the environment level value will be used.

Cache Corp Flag

```
DEFINE CACHE_ENABLED='true';
```

2.2 Use

The Cache Corporation flag can be retrieved from the configuration database by using the following methods in the CacheUtil class:

Method	Input	Output	Description
getCacheConnectorConfig	NA	CacheConnectorConfiguration	Returns the DEFAULT Cache Corporation flag. Not corp specific.
getCacheConnectorConfig	Corporation number	CacheConnectorConfiguration	Returns the Cache Corporation flag if exists at the Corporation level, if not found the DEFAULT value configured will be returned.

This flag can be updated at runtime via [Admin Services](#) (see page 16).

3 B. Cache Service Status

The Cache Service Status is an indicator used to configure how each one will be started. It can be always-on at startup, enabled only during stand-in mode or disabled temporarily.

- Defined in the API service SQL configuration.
- Configuration category: V2.SERVICE.DEFINITION
- Value is defined in the '<backside Transport>' section, XML tag name '<cacheEnabled>'.

There are four valid values that define the Cache service behavior. Valid values include:

- 'A' (Always-on): indicates that the Cache service will always be called (e.g. at start-up, during batch, outages etc..). Default value for all inquiry services.
- 'E' (Enabled): for update services. The Cache service will only be called if defined with an 'E' status AND the Cache stand-in mode flag is also TRUE.
- 'D' (Disabled): this is a status used when a Cache service is disabled temporarily via [Admin Services](#) (see page 16) or service configuration changes.
- 'N' (None): default value for non-cache services.

3.1 Configuration

The status is defined in the XML Proxy by adding a new attribute 'cached' to the <ServiceLinkData> tag. Example to set the BLK cache service status to 'Enabled':

Set Cache Service Status

```
<BGProxyMetadata>
  <Name>BCSVSBLK</Name>
  <V2XMLHdrVersion>2.0.0</V2XMLHdrVersion>
  <Version release="February 2023">v1.0.0</Version>
  <ServiceLinkData requestCode="BLK" description="Change Block/Reclass" tokenised="
true" cached="E" />
  <Request>
    <Header>
      <CBName>BCSVLRCV</CBName>
      <CBRedefines>>false</CBRedefines>
    </Header>
    <Data>
      <CBName>BCSVWBLK</CBName>
      <CBRootField>SVL-BLK-DATA-BLOCK</CBRootField>
    </Data>
  </Request>
</Response>
```

```

    <Header>
      <CBName>BCSVLSND</CBName>
    </Header>
  </Response>
</BGProxyMetadata>

```

3.2 Use

The Cache Service Status valid values are defined as an enumeration in the CacheConstants interface.

At the Infinity Connect application startup, each API service configuration along with its Cache Service status is loaded in-memory. When an API is called the respective Cache Service status is retrieved and evaluated along with the other flags before sending the MQ request to the back-end (P1C or Cache module). This logic will be implemented in the BASEGlobalBacksideAdapter.

Method	Input	Output	Description
getCacheServiceStatus	AdapterParms	CacheConstants.SERVICE_STATUS	Reads the service configuration and returns the Cache Service Status value. If configuration is not found or Cache Service Status value is empty/invalid a SERVICE_STATUS.NONE (N) is returned.

4 C. Cache Stand-In Flag

This flag is automatically enabled (TRUE) by IC processing when NRT batch commands (QCMD) or XPH Ping Service extended processing system status values indicate the cache application is to be used during extended processing phases, release implementations and planned IPLs (i.e., when P1C mainframe functionality is temporarily unavailable).

The logic to support Cache Stand-In Flag will be implemented in the BASEGlobalBacksideAdapter.

Method	Input	Output	Description
isCacheStandInMode	Corporation Number	boolean	Based on the QCMD batch phase and XPH extended system status, see tables 1 and 2 below, true will be returned if P1C is down/unavailable or false if available.

1) P1C Scheduled Daily Batch Outages

The table below is from BD (v1-03) and shows:

- When Cache can Stand-in during select P1C Extended phases of the batch cycle (see Orange rows).
- When IC Cache processing will automatically set the Service Cache Replica Stand-in Flag.
- When Bulkload can release Cache SAF Pending requests to P1C.

P1C		Cache Stand-in / Send SAF Requests				
P1C Daily Batch Phase	Triggered by	Last P1C NRT Batch Command (QCMD)		XPH Extended System Status Response	CACHE Stand-in (Y/N)	Send SAF Requests (Y/N)
Normal	N/A	N/A		Spaces	N	Y
End of Day	P1C Batch process (P#2BB640)	BATCHBEG	or	'TE' Transition to Extended	Y	N
Extended Limited	N/A	N/A		'EL' Extended Limited	N	Y

P1C		Cache Stand-in / Send SAF Requests				
P1C Daily Batch Phase	Triggered by	Last P1C NRT Batch Command (QCMD)		XPH Extended System Status Response	CACHE Stand-in (Y/N)	Send SAF Requests (Y/N)
Extended Normal	N/A	N/A		'E ' Extended Normal	N	Y
Greyhole Authorisation processing	N/A	BATCHAUTH		'E ' Extended Normal <i>Note: This marks the start of P1C Greyhole processing and when the Cache application will start to store NRT Authorisation messages that will be applied in the Cache once batch has completed.</i>	N	Y
Beginning of Day	Interactive Processing Request Manager (BCOS81) BEGOFDAY	TRANTOCON (new)	or	'TC' Transition to Consolidated	Y	N
Initiate Authorisations	N/A	CONLIM		'IA' Init Auth	N	Y
Consolidated Limited	N/A	N/A		'CL' Consolidated Limited	N	Y
Consolidated Normal	N/A	BATCHEND		'C ' Consolidated Normal	N	Y
Consolidation to Normal	Interactive Processing Request Manager (BCOS81) CONTONOR	TRANTONOR (new)	or	'TN' Transition to Normal	Y	N

P1C		Cache Stand-in / Send SAF Requests				
P1C Daily Batch Phase	Triggered by	Last P1C NRT Batch Command (QCMD)		XPH Extended System Status Response	CACHE Stand-in (Y/N)	Send SAF Requests (Y/N)
Normal	N/A	N/A		Spaces	N	Y
Extended System Status Abend	N/A	N/A		'XX' Abend – can be applicable throughout all P1C extended processing phases above	Y	N

4.1 QCMD Batch Phase

P1C QCMD NRT messages communicate the batch status to all the IC instances via MQ Topic. The MQ message is built by P1C using the PCRMDCMD copybook and the batch phase is sent in the MDCMD_COMMAND field. The QCMDBCADDService in the xpressng-b2k-cache-services-pcrmdext project handles these messages and batch phases are read/stored in-memory using the following methods in the CacheUtil class:

Method	Input	Output	Description
setPCRMDCMD	PCRMDCMD_Record	NA	Saves the QCMD MQ record in-memory and indexed by corporation. QCMD messages are expected to be sent with corporation 00. The values are stored using the IC Cache Manager and key 'PCRMDCMD'. Only one batch phase (latest QCMD message) is stored in memory at a time.
getPCRMDCMD	NA	PCRMDCMD_Record	Returns the QCMD MQ record from memory using default corporation 00.

Method	Input	Output	Description
getPCRMDCMD	Corporation Number	PCRMDCMD_Reco rd	Returns the QCMD MQ record from memory using a specific corporation. If no record is found, another lookup is done but with the default corporation 00. If no record found a warning is logged and null returned.

4.2 XPH Extended System Status

The XPH Service Link will be enhanced to return the batch extended system status indicator (field SVL-XPH-S-BATCH-STATUS).

PCSVWXPX Sender Copybook			
000100*N	CC7836		PCSVWXPX
000200*****			PCSVWXPX
000300*	P C S V W X P H	C O P Y B O O K	*PCSVWXPX
000400*			*PCSVWXPX
000500*	THIS COPYBOOK CONTAINS DATA ELEMENTS THAT FOR THE XPH		*PCSVWXPX
000600*	REQUEST (PING HOST). THIS WILL ONLY CONTAIN RESPONSE DATA.		*PCSVWXPX
000700*			*PCSVWXPX
000800*****			PCSVWXPX
000900*			PCSVWXPX
001000*			PCSVWXPX
001100 01	SVL-XPH-S-DATA-BLOCK.		PCSVWXPX
001200 05	SVL-XPH-S-BATCH-STATUS	PIC X(02).	PCSVWXPX
001300 88	SVL-XPH-S-BS-NORMAL	VALUE ' '.	PCSVWXPX
001400 88	SVL-XPH-S-BS-TRAN-TO-EXT	VALUE 'TE'.	PCSVWXPX
001500 88	SVL-XPH-S-BS-EXTEND-LIM	VALUE 'EL'.	PCSVWXPX
001600 88	SVL-XPH-S-BS-EXTEND-NOR	VALUE 'E '.	PCSVWXPX
001700 88	SVL-XPH-S-BS-TRAN-TO-CONSOL	VALUE 'TC'.	PCSVWXPX
001800 88	SVL-XPH-S-BS-INIT-AUTH	VALUE 'IA'.	PCSVWXPX
001900 88	SVL-XPH-S-BS-CONLIM	VALUE 'CL'.	PCSVWXPX
002000 88	SVL-XPH-S-BS-CON-TO-NOR	VALUE 'C '.	PCSVWXPX
002100 88	SVL-XPH-S-BS-TRAN-TO-NOR	VALUE 'TN'.	PCSVWXPX
002200 88	SVL-XPH-S-BS-ABENDED	VALUE 'XX'.	PCSVWXPX
002300*			PCSVWXPX
999700*****			PCSVWXPX
999800*	END OF COPYBOOK		*PCSVWXPX
999900*****			PCSVWXPX

The Infinity Connect XPH API is an active health check service used by the load balancer. Successful XPH calls store the SVL-XPH-S-BATCH-STATUS and read it from memory using the following CacheUtil class methods:

Method	Input	Output	Description
setXPHExtendedSystemStatus	BCSVSXPHE_Record	NA	Saves the batch XPH status value in-memory. This is mapped and stored as an enumeration XPH_BATCH_STATUS. Status is stored in the IC Cache Manager with key 'PCRMDCMD'. Only one XPH status (latest successful XPH message) is stored in memory at a time.
getXPHExtendedSystemStatus	NA	XPH_BATCH_STATUS	Returns an XPH batch status is found in memory or else a warning is logged and returns by default XPH_BATCH_STATUS.SPACES.

2) Planned P1C outage (i.e. P1C Release Implementation and IPLs)

The table below is from BD (v1-03) and shows:

- When Cache can Stand-in during a planned P1C outage, i.e. P1C Release Implementation and IPLs (see Orange rows).
- When IC Cache processing will automatically set the Service Cache Replica Stand-in Flag (see Section 3.3.2.13).
- When Bulkload can release Cache SAF Pending requests to P1C after the outage when notified that P1C is available to receive service requests. This notification requires Bulkload to use both the Last P1C NRT Batch Command (QCMD) and XPH Extended System Status Response as below.

P1C		Cache Stand-in / Send SAF Requests				
P1C Daily Batch Phase	Triggered by	Last P1C NRT Batch Command (QCMD)		XPH Extended System Status Response	CACHE Stand-in (Y/N)	Bulkload Send SAF Requests (Y/N)
Start of Outage	P1C Batch process (new)	SIGNOFF		N/A	Y	N
End of Outage	P1C Batch process (new)	SIGN ON	and	Spaces (Normal)	N	Y

3) Unplanned outage (IC timeout/connection issues) - TODO

IC and related services are supported by multiple servers.

Cache may also Stand-in during an unplanned outage on a specific server(s):

- When Internal Ping Host (XPH) Service Link or any other Service Link request is unable to be sent to P1C due to timeout/connection issues; and
- There hasn't been an XPH, or other Service Link request successfully sent to P1C on that Server for 'x' timeframe. Where 'x' will be initially defined as 5 seconds by default.

Note: in a multiple server environment, this could result in one or more servers being in Cache Stand-in while others are not.

When the timeout/connection issues are resolved on the specific server, the Cache will no longer Stand-in unless it is required to as per the criteria in items 1) and 2) above.

5 Admin Services

Infinity Connect Admin services allow reading or updating configuration values used by the application at runtime without the need to restart the server.

These can be accessed through the following URL `https://{IP}:{PORT}/xpress/pox/v2rqst` and require system supervisor credentials to run.

The following Admin service were built for CC7836 and are use do manage the Cache Corporation Flag and any Cache Service Status.

5.1 Corporation Flag Inquiry (ConfigCache v1.0.0)

Allows to read the current Cache Corporation Flag.

Monitor XML Tags required:

XML Tag	Required	Value
<Name>	Y	ConfigCache
<Feld>	N	Corporation number when flag is defined at the corporation level otherwise use DEFAULT or do not send the tag to return the value set at the application level

Request/Response samples:

ConfigCache - Request
<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:fis=" FISNgSystemMonitorRqst" xmlns:fis1="FisCommonRqstHdr" xmlns:fis2="FISNgCacheMonitorRq st" xmlns:fis3="FISNgConfigMonitorRqst" xmlns:fis4="FISNgConfigMonitorRqst"> <soapenv:Header/> <soapenv:Body> <fis:NgSystemSupervisorRqst xmlns:fis="FISNgSystemSupervisorRqst"> <fis1:RqstHdr> <fis1:RqstHdrVer>2.0.0</fis1:RqstHdrVer> <fis1:MsgUid>c399df90-cfe2-456b-856e-a1151baa1c2c</fis1:MsgUid> <fis1:SrcId>Admin</fis1:SrcId> <fis1:LclPref>en-US</fis1:LclPref> <fis1:Sec> <fis1:BasicAuthen> <fis1:UsrId>JMXAdmin</fis1:UsrId> </fis1:BasicAuthen> </fis1:Sec> </fis1:RqstHdr> </fis:NgSystemSupervisorRqst> </soapenv:Body> </soapenv:Envelope> </pre>

```

        <fis1:Pswrd>NG@admin</fis1:Pswrd>
    </fis1:BasicAuthen>
</fis1:Sec>
<fis1:ServPrmtrsLst>
    <fis1:ServPrmtrs>
        <fis1:FeId>LOCAL</fis1:FeId>
        <fis1:ServId>SystemSupervisor</fis1:ServId>
        <fis1:ServVer>v1.0.0</fis1:ServVer>
        <fis1:ApplId>XpressNG</fis1:ApplId>
    </fis1:ServPrmtrs>
</fis1:ServPrmtrsLst>
</fis1:RqstHdr>
<fis:RqstBody>
    <fis:MonitorLst>
        <fis:Monitor>
            <Name>ConfigCache</Name>
            <FeId>30</Name> <!-- Optional tag used if flag is defined at the
corporation level, DEFAULT used if not sent -->
        </fis:Monitor>
    </fis:MonitorLst>
</fis:RqstBody>
</fis:NgSystemSupervisorRqst>
</soapenv:Body>
</soapenv:Envelope>

```

ConfigCache - Valid Response

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <NgSystemSupervisorRsp xmlns="FISNgSystemSupervisorRsp">
      <RspHdr xmlns="FisCommonRspHdr">
        <RspHdrVer>2.0.0</RspHdrVer>
        <RtnCde>0</RtnCde>
        <MsgUuid>f71982ee-ca00-487f-ab23-c89e498a6f95</MsgUuid>
        <Lcl>en-US</Lcl>
      </RspHdr>
      <RspBody>
        <MonitorLst>
          <Monitor>
            <Name>ConfigCache</Name>
            <Result>Success</Result>
            <Response>
              <ConfigMonitorLst xmlns="FISNgConfigMonitorRsp">
                <ConfigMonitor>
                  <ConfigName>BASEGLOBAL.CACHE.CONNECTOR</ConfigName>
                  <CacheInstance>amo8rnabxngas03.cust.au~8nk/
nbovB0WbBYLZ80hLoAd7ujFP5J/z5MU6tx7e9Y8=</CacheInstance>
                  <CacheHostname>amo8rnabxngas03.cust.au</CacheHostname>

```

```

        <CacheIp>10.56.12.141</CacheIp>
        <FeId>DEFAULT</FeId>
        <CacheEnabled>true</CacheEnabled>
    </ConfigMonitor>
</ConfigMonitorLst>
</Response>
</Monitor>
</MonitorLst>
</RspBody>
</NgSystemSupervisorRsp>
</soapenv:Body>
</soapenv:Envelope>

```

ConfigCache - Invalid Response (flag not defined at the corporation level)

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <NgSystemSupervisorRsp xmlns="FISNgSystemSupervisorRsp">
      <RspHdr xmlns="FisCommonRspHdr">
        <RspHdrVer>2.0.0</RspHdrVer>
        <RtnCde>-1</RtnCde>
        <MsgUuid>9c674108-c0c0-46b6-9295-ee312c25d000</MsgUuid>
        <Lcl>en-US</Lcl>
      </RspHdr>
      <RspBody>
        <MonitorLst>
          <Monitor>
            <Name>ConfigCache</Name>
            <Result>Failed</Result>
            <Response>Unable to process message content Cache configuration not
found for corporation=30</Response>
          </Monitor>
        </MonitorLst>
      </RspBody>
    </NgSystemSupervisorRsp>
  </soapenv:Body>
</soapenv:Envelope>

```

5.2 Cache Service Status Flag (ConfigCacheSvc v1.0.0)

Allows to inquiry all the Cache Services Status flags.

Monitor XML Tags required:

XML Tag	Required	Value
<Name>	Y	ConfigCacheSvc
<Feld>	N	Corporation number when flag is defined at the corporation level otherwise use DEFAULT or do not send the tag to return the value set at the application level

Request/Response samples:**ConfigCacheSvc - Request**

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:fis="
FISNgSystemMonitorRqst" xmlns:fis1="FisCommonRqstHdr" xmlns:fis2="FISNgCacheMonitorRq
st" xmlns:fis3="FISNgConfigMonitorRqst" xmlns:fis4="FISNgConfigMonitorRqst">
  <soapenv:Header/>
  <soapenv:Body>
    <fis:NgSystemSupervisorRqst xmlns:fis="FISNgSystemSupervisorRqst">
      <fis1:RqstHdr>
        <fis1:RqstHdrVer>2.0.0</fis1:RqstHdrVer>
        <fis1:MsgUuid>c399df90-cfe2-456b-856e-a1151baa1c2c</fis1:MsgUuid>
        <fis1:SrcId>SEP123</fis1:SrcId>
        <fis1:LclPref>en-US</fis1:LclPref>
        <fis1:Sec>
          <fis1:BasicAuthen>
            <fis1:UsrId>JMXAdmin</fis1:UsrId>
            <fis1:Pswrd>NG@dmin</fis1:Pswrd>
          </fis1:BasicAuthen>
        </fis1:Sec>
        <fis1:ServPrmtrsLst>
          <fis1:ServPrmtrs>
            <fis1:FeId>LOCAL</fis1:FeId>
            <fis1:ServId>SystemSupervisor</fis1:ServId>
            <fis1:ServVer>v1.0.0</fis1:ServVer>
            <fis1:ApplId>XpressNG</fis1:ApplId>
          </fis1:ServPrmtrs>
        </fis1:ServPrmtrsLst>
      </fis1:RqstHdr>
      <fis:RqstBody>
        <fis:MonitorLst>
          <fis:Monitor>
            <Name>ConfigCacheSvc</Name>
            <FeId>DEFAULT</FeId> <!-- Optional tag used if flag is defined at
the corporation level, DEFAULT used if not sent -->
          </fis:Monitor>
        </fis:MonitorLst>
      </fis:RqstBody>
    </fis:NgSystemSupervisorRqst>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    </fis:RqstBody>
  </fis:NgSystemSupervisorRqst>
</soapenv:Body>
</soapenv:Envelope>

```

ConfigCacheSvc - Valid Response

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <NgSystemSupervisorRsp xmlns="FISNgSystemSupervisorRsp">
      <RspHdr xmlns="FisCommonRspHdr">
        <RspHdrVer>2.0.0</RspHdrVer>
        <RtnCde>0</RtnCde>
        <MsgUuid>102c9a57-ed9b-43b9-ad9b-b2fa42b3fd47</MsgUuid>
        <Lcl>en-US</Lcl>
      </RspHdr>
      <RspBody>
        <MonitorLst>
          <Monitor>
            <Name>ConfigCacheSvc</Name>
            <Result>Success</Result>
            <Response>
              <ConfigMonitorLst xmlns="FISNgConfigMonitorRsp">
                <ConfigMonitor>
                  <ConfigName>V2.SERVICE.DEFINITION</ConfigName>
                  <APILst>
                    <API>
                      <FeId>DEFAULT</FeId>
                      <RequestCode>CDU</RequestCode>
                      <Version>v1.2.0</Version>
                      <CacheEnabled>E</CacheEnabled>
                    </API>
                    <API>
                      <FeId>DEFAULT</FeId>
                      <RequestCode>GMA</RequestCode>
                      <Version>v1.0.0</Version>
                      <CacheEnabled>A</CacheEnabled>
                    </API>
                    <API>
                      <FeId>DEFAULT</FeId>
                      <RequestCode>XT3</RequestCode>
                      <Version>v3.1.0</Version>
                      <CacheEnabled>A</CacheEnabled>
                    </API>
                    <API>
                      <FeId>DEFAULT</FeId>
                      <RequestCode>XTA</RequestCode>
                      <Version>v1.1.0</Version>
                    </API>
                  </APILst>
                </ConfigMonitor>
              </ConfigMonitorLst>
            </Response>
          </Monitor>
        </MonitorLst>
      </RspBody>
    </NgSystemSupervisorRsp>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>XTW</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>XTW</RequestCode>
    <Version>v2.0.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>NAA</RequestCode>
    <Version>v1.2.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>NAA</RequestCode>
    <Version>v2.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>NAC</RequestCode>
    <Version>v1.2.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>NAC</RequestCode>
    <Version>v2.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CLC</RequestCode>
    <Version>v1.2.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>ODO</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>

```

```

    <RequestCode>BLK</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>COC</RequestCode>
    <Version>v2.5.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>USR</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>MCU</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>SRH</RequestCode>
    <Version>v2.1.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>SRX</RequestCode>
    <Version>v1.2.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>ONP</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>ONP</RequestCode>
    <Version>v2.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>ATU</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>

```

```

<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>PRO</RequestCode>
  <Version>v1.1.0</Version>
  <CacheEnabled>E</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>P2P</RequestCode>
  <Version>v1.1.0</Version>
  <CacheEnabled>E</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>GCC</RequestCode>
  <Version>v1.1.0</Version>
  <CacheEnabled>A</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>GCC</RequestCode>
  <Version>v2.0.0</Version>
  <CacheEnabled>A</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>GCD</RequestCode>
  <Version>v1.0.0</Version>
  <CacheEnabled>A</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>MTS</RequestCode>
  <Version>v1.0.0</Version>
  <CacheEnabled>E</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>LSR</RequestCode>
  <Version>v1.0.0</Version>
  <CacheEnabled>E</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>LSR</RequestCode>
  <Version>v2.0.0</Version>
  <CacheEnabled>E</CacheEnabled>
</API>
<API>
  <FeId>DEFAULT</FeId>
  <RequestCode>CAH</RequestCode>
  <Version>v1.1.0</Version>

```

```

    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CAS</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>D</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CBI</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CBO</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CBT</RequestCode>
    <Version>v1.2.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CCA</RequestCode>
    <Version>v2.5.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CCI</RequestCode>
    <Version>v2.4.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>SCB</RequestCode>
    <Version>v1.0.0</Version>
    <CacheEnabled>A</CacheEnabled>
</API>
<API>
    <FeId>DEFAULT</FeId>
    <RequestCode>CCU</RequestCode>
    <Version>v2.5.0</Version>
    <CacheEnabled>E</CacheEnabled>
</API>
</APIList>
</ConfigMonitor>

```

```

        </ConfigMonitorLst>
    </Response>
</Monitor>
</MonitorLst>
</RspBody>
</NgSystemSupervisorRsp>
</soapenv:Body>
</soapenv:Envelope>

```

ConfigCacheSvc - Invalid Response (corporation does not exist. Empty list of Cache services returned)

Request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:fis="
FISNgSystemMonitorRqst" xmlns:fis1="FisCommonRqstHdr" xmlns:fis2="FISNgCacheMonitorRq
st" xmlns:fis3="FISNgConfigMonitorRqst" xmlns:fis4="FISNgConfigMonitorRqst">
  <soapenv:Header/>
  <soapenv:Body>
    <fis:NgSystemSupervisorRqst xmlns:fis="FISNgSystemSupervisorRqst">
      <fis1:RqstHdr>
        <fis1:RqstHdrVer>2.0.0</fis1:RqstHdrVer>
        <fis1:MsgUuid>c399df90-cfe2-456b-856e-a1151baa1c2c</fis1:MsgUuid>
        <fis1:SrcId>SEP123</fis1:SrcId>
        <fis1:LclPref>en-US</fis1:LclPref>
        <fis1:Sec>
          <fis1:BasicAuthen>
            <fis1:UsrId>JMXAdmin</fis1:UsrId>
            <fis1:Pswrd>NG@dmn</fis1:Pswrd>
          </fis1:BasicAuthen>
        </fis1:Sec>
        <fis1:ServPrmtrsLst>
          <fis1:ServPrmtrs>
            <fis1:FeId>LOCAL</fis1:FeId>
            <fis1:ServId>SystemSupervisor</fis1:ServId>
            <fis1:ServVer>v1.0.0</fis1:ServVer>
            <fis1:ApplId>XpressNG</fis1:ApplId>
          </fis1:ServPrmtrs>
        </fis1:ServPrmtrsLst>
      </fis1:RqstHdr>
      <fis:RqstBody>
        <fis:MonitorLst>
          <fis:Monitor>
            <Name>ConfigCacheSvc</Name>
            <FeId>60</FeId>
          </fis:Monitor>
        </fis:MonitorLst>
      </fis:RqstBody>
    </fis:NgSystemSupervisorRqst>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    </soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <NgSystemSupervisorRsp xmlns="FISNgSystemSupervisorRsp">
      <RspHdr xmlns="FisCommonRspHdr">
        <RspHdrVer>2.0.0</RspHdrVer>
        <RtnCde>0</RtnCde>
        <MsgUuid>c3de0e92-e47c-476d-a1e2-58ba22f88c09</MsgUuid>
        <Lcl>en-US</Lcl>
      </RspHdr>
      <RspBody>
        <MonitorLst>
          <Monitor>
            <Name>ConfigCacheSvc</Name>
            <Result>Success</Result>
            <Response>
              <ConfigMonitorLst xmlns="FISNgConfigMonitorRsp">
                <ConfigMonitor>
                  <ConfigName>V2.SERVICE.DEFINITION</ConfigName>
                  <APILst/>
                </ConfigMonitor>
              </ConfigMonitorLst>
            </Response>
          </Monitor>
        </MonitorLst>
      </RspBody>
    </NgSystemSupervisorRsp>
  </soapenv:Body>
</soapenv:Envelope>

```